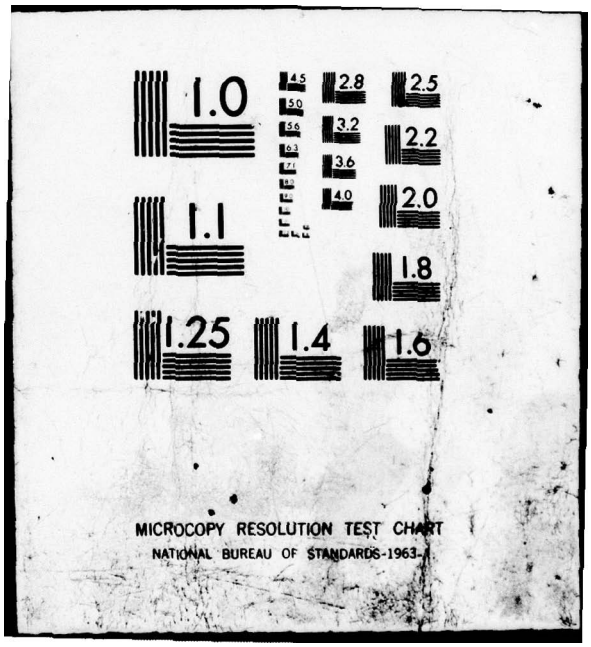
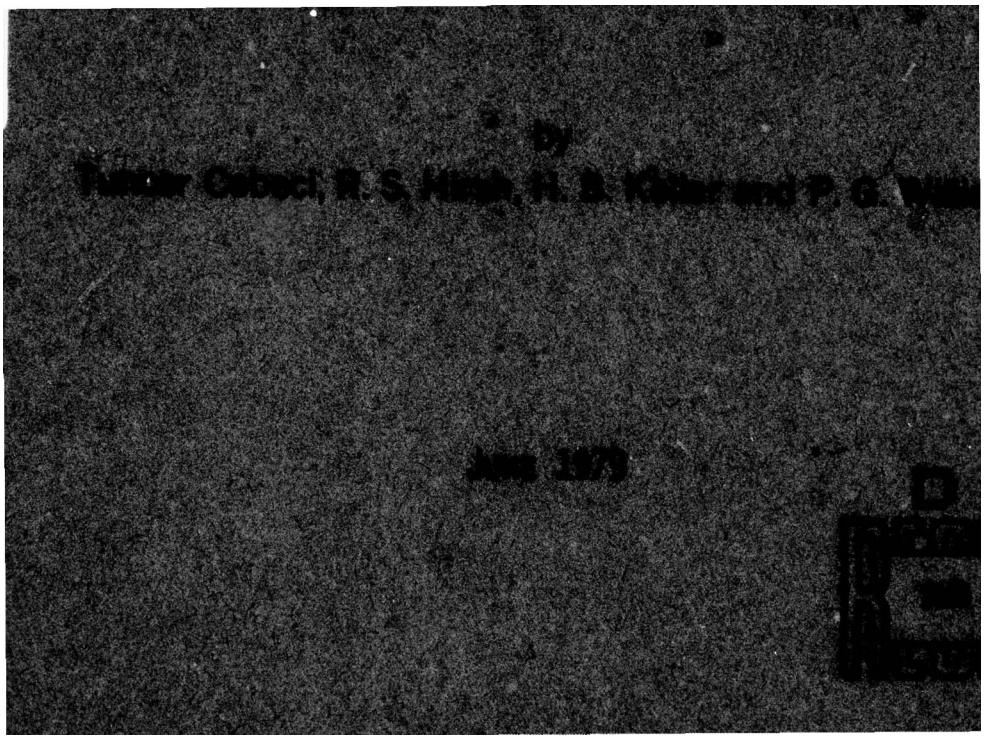
AD-A073 236 DOUGLAS AIRCRAFT CO LONG BEACH CA F/G 12/1
STUDIES OF NUMERICAL METHODS FOR THE PLANE NAVIER-STOKES EQUATI--ETC(U)
JUN 79 T CEBECI, R S HIRSH, H B KELLER N00019-78-C-613
UNCLASSIFIED MDC-J8525 NL

| OF |
AD
A073236

END
DATE
FILMED
9-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



Copy number

Report number (14) MDC J8525

(6) STUDIES OF NUMERICAL METHODS FOR THE PLANE NAVIER-STOKES EQUATIONS

Revision date

Revision letter

Issue date (11) June 1979

Contract number (15) N00019-78-C-0613

(10) Prepared by: Tuncer Cebeci, R. S. Hirsh, H. B. Keller and P. G. Williams

(12) 54 p.

Approved by:

F. T. Lynch
Branch Chief
Research and Development
Aerodynamics Subdivision

R. B. Harris
Director – Aerodynamics

DOUGLAS AIRCRAFT COMPANY,
Long Beach, CA
MCDONNELL DOUGLAS CORPORATION

116 400

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Basic Problems

The plane, steady, incompressible, Navier-Stokes equations have been subjected to a variety of numerical attacks. At the present state of the art there is no particular method that is clearly recognized as the "best" way to solve these equations. The difficulties are varied and perhaps the fact that we cannot easily list and order them in importance points up the largest difficulty of all. To make some reasonable start on studies of numerical methods for the plane, steady, Navier-Stokes equations we must first classify, at least in some rough sense, the types of problems that are of interest.

The first basic classification concerns the domain (i.e. geometry) over which the flow is of interest and the boundary conditions are to be imposed. In all "external" flows, or domains extending to infinity, we must be able to give proper conditions on some artificial boundary in the fluid, in order to reduce the problem to one in a finite domain. We do not attempt to study this problem in our present work. Instead we use test problems in which boundary conditions over simple finite domains are "reasonably" well defined. For example, we consider the driven cavity and the inlet region of a channel. Even here, of course, there are open questions. But they do not dominate the problem as, say, the flow at large radius does in the flow about a cylinder.

The next basic difficulty concerns the Reynolds number, R, or rather the range of R over which solutions are sought. For sufficiently large R there will be thin boundary layers in the flow, say of width δ. Thus the net spacing and numerical method must be able to resolve variations over

lengths of the order of $\delta/10$ say. We shall not attempt to resolve this open question either. But our methods must be capable of working at large R if we can use appropriate fine nets. Also, we point out that most calculations are desired not for a fixed R value but rather over some interval, say $R_0 < R < R_F$. This fact can play a crucial role in developing efficient numerical methods.

Suppose then that we have a simple domain (a rectangle for example) on all of whose boundaries the "correct" boundary conditions are known. Further, steady solutions are desired for a "moderate" range of R values. All numerical attempts to solve such problems are iterative. The literature on this subject does not always stress these facts but it is clear that the dominant concerns are now: (a) to insure convergence of whatever iteration scheme is used, (b) to accelerate convergence assuming it can be attained, (c) to insure that the converged iterates accurately approximate a physical solution, (d) to insure that this solution is the desired one (for steady solutions may not be unique). Of course, all these concerns are interrelated and may not be resolved independently of each other.

1.2 Basic Methods

In the present report and in our most recent work we have concentrated on three basic formulations: primitive variables in Section 2.0, vorticity-stream function in Section 3.0, and stream function-biharmonic formulation, in Section 4.0. With each such formulation there are a variety of difference equations that could be used and then there are a large number of iteration schemes that could be employed to solve these difference equations. The ultimate goal, of course, is to find the best combination of all these techniques.

We have discontinued, for the present, our earlier primitive variable-splitting technique studies[1]. They have some very attractive features but, as we had used them, they seemed quite sensitive to the boundary treatment. Also, their convergence rates did not compare to that of our new biharmonic method.

The primitive variable methods of Section 2.0 are based on the reportedly successful techniques of Spalding[2] and his coworkers. Basically, ADI methods are used to compute the velocity from the momentum equations and then these velocities and the pressure are adjusted to satisfy continuity. This latter procedure is closely equivalent to solving a Poisson equation for the pressure. This is also done by an inner loop of ADI. Careful tuning of this method seems to be required and adjustments of various kinds are made when it is applied to different problems. It is not clear that any analysis of convergence rates or order of accuracy can be done for this complicated method.

The vorticity-stream function formulation of Section 3.0 is classical for plane steady viscous flows. With more recent developments in numerical analysis, it should be possible to make this formulation much more efficient than it has been in the past. Thus we use ADI on the vorticity equation, suitably linearized, and SOR on the Poisson equation (vorticity definition). There are the usual troubles with boundary conditions in this method since vorticity is not known on the boundaries and is in fact generated there. It seems clear that the use of fast Poisson solvers could greatly enhance the efficiency of these procedures; we hope to investigate this in the future.

Finally, in Section 4.0 we use the biharmonic or fourth-order formulation in terms of a stream-function (i.e. eliminate the vorticity). Now there are no problems with the proper boundary conditions (on rigid walls,

for instance). Also, there are not coupled systems of differential equations to be solved. Newton's method and modifications of it, in particular the Newton-Chord method, are extremely fast for solving the finite difference equivalents of this formulation. The main drawbacks with this method are the possibly large storage requirements and the time it takes to get the LU-decomposition of the large band-structured coefficient matrix that occurs. But tentative tests indicate that this scheme can be as fast as the others and works for a far greater range of Reynolds numbers.

2.0 PRIMITIVE VARIABLE FORMULATION

2.1 Equations and Grid

Of the few methods which solve the incompressible Navier-Stokes equations in primitive variables, the most successful one derives from the work of Chorin[3], Amsden and Harlow[4], and Spalding[2]. This entire group can be described as being "pressure corrector" methods. The meaning of this phrase will become clear in the detailed description which follows.

The most familiar of these methods are the ones which have been developed at Imperial College. The latest in the series of computer programs which utilize this pressure corrector scheme is known as TEACH. Now, although the basic method is the pressure corrector approach, numerous computational deviations from a straightforward application of the method have been incorporated into the TEACH code to render it useful for solving practical problems. These devices range from the incorporation of upwind differencing, and different under-relaxation parameters for different variables, to the inclusion of "false source" terms in the equations[5] in order to enhance convergence, and the use of a block correction procedure[6] to correct the difference equations for discretization errors. All these tend to obscure the basic properties of the pressure corrector method itself and hence make comparisons with other methods difficult, so the procedure described below is the basic concept of the pressure corrector method, applied to various test problems, but bereft of all computational devices.

Previous investigations have provided related information concerning implementation of the method (see Spalding[2] or Roache[7] for example) but it is useful to have a complete, detailed, development in one self-contained work, and much of the interpretation and some details which are included here are new.

The equations to be solved are the suitability nondimensional incompressible form of the Navier-Stokes equations, i.e.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{R}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2.1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{R}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{2.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.3}$$

In these equations the time derivative is regarded as a means of approaching the asymptotic steady state, and a consistent time development of the equations will not be considered. The TEACH method considers the steady equations, and solves this elliptic system by relaxation methods. The conservation form of these equations can be shown to possess the conservative property[8], hence conserve mass in the global sense when cast in finite-difference form, so instead of eqs. (2.1) and (2.2), the following equations are used:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(u^2) + \frac{\partial}{\partial y}(uv) = -\frac{\partial p}{\partial x} + \frac{1}{R}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2.4}$$

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}(uv) + \frac{\partial}{\partial y}(v^2) = -\frac{\partial p}{\partial y} + \frac{1}{R}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{2.5}$$

The finite-difference form of these equations is solved on the staggered grid depicted in fig. 1. Arguments as to the physical basis for this grid have been made, and it has become standard practice for all primitive variable solution techniques (including finite-element methods[9]). However, it should be noted that the original pressure corrector method of Chorin did not use this grid, but the standard one with all variables defined at the same point.

Note the location of the boundaries of the solution domain and the position of each variable relative to the grid and boundaries. On this grid the interior values of the pressure, $p_{i,j}$, are defined on $2 \leq i \leq IMAX$ and $2 \leq j \leq JMAX$. The values along grid lines 1 or MAX+1 must be

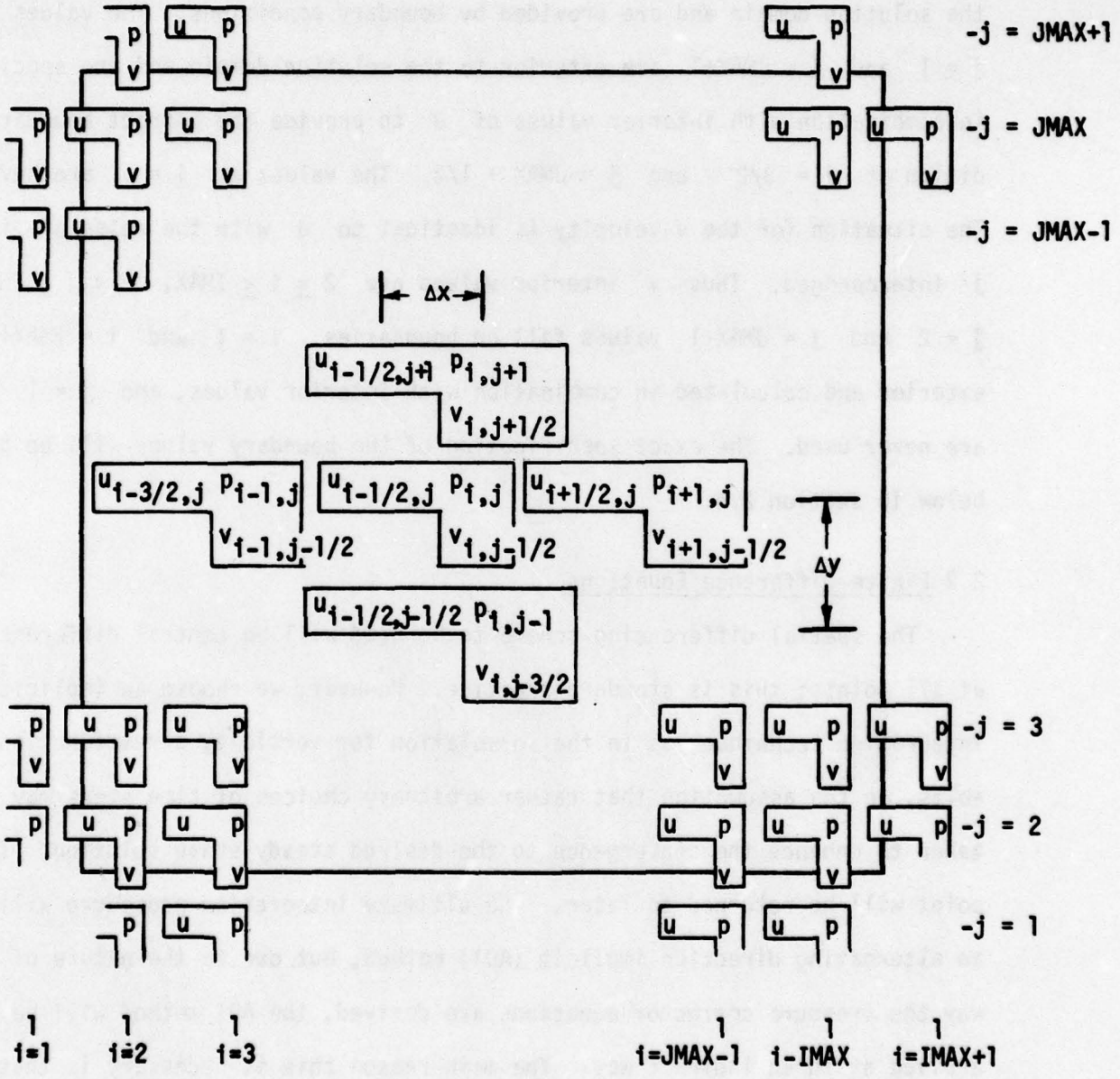

-j = JMAX+1
-j = JMAX
-j = JMAX-1
Δx
$u_{i-1/2,j+1}$ $p_{i,j+1}$ $v_{i,j+1/2}$
$u_{i-3/2,j}$ $p_{i-1,j}$ $v_{i-1,j-1/2}$
$u_{i-1/2,j}$ $p_{i,j}$ $v_{i,j-1/2}$
$u_{i+1/2,j}$ $p_{i+1,j}$ $v_{i+1,j-1/2}$
Δy
$u_{i-1/2,j-1/2}$ $p_{i,j-1}$ $v_{i,j-3/2}$
-j = 3
-j = 2
-j = 1
i=1
i=2
i=3
i=JMAX-1
i=IMAX
i=IMAX+1

Figure 1

determined by boundary conditions, or, as will be seen later, other means. The interior values of the u-velocity are defined on $3 \leq i \leq IMAX$, $2 \leq j \leq JMAX$. The values at $i = 2$ and $i = IMAX+1$ fall directly on the boundary lines of the solution domain and are provided by boundary conditions. The values at $j = 1$ and $j = JMAX+1$ are exterior to the solution domain and are specified in combination with interior values of u to provide the correct boundary condition at $j = 3/2$ and $j = JMAX + 1/2$. The values at $i = 1$ are never used. The situation for the v-velocity is identical to u with the roles of i and j interchanged. Thus v interior values are $2 \leq i \leq IMAX$, $3 \leq j \leq JMAX$; $j = 2$ and $j = JMAX+1$ values fall on boundaries. $i = 1$ and $i = IMAX+1$ are exterior and calculated in combination with interior values, and $j = 1$ values are never used. The exact specification of the boundary values will be given below in section 2.4.

2.2 Finite-Difference Equations

The spatial differencing scheme to be used will be central differencing at all points; this is standard practice. However, we choose an implicit integration technique, as in the formulation for vorticity streamfunction variables, on the assumption that rather arbitrary choices of time steps may be taken to enhance the convergence to the desired steady-state solution. This point will be returned to later. The ultimate integration procedure will be an alternating direction implicit (ADI) method, but due to the nature of the way the pressure corrector equations are derived, the ADI method will be arrived at in an indirect way. The main reason this is necessary is that the pressure corrector equations are derived from considering the linearized difference equations and not the original differential equations.

To derive the final form of the equations, consider the fully-implicit differencing of equations (2.4) and (2.5). The x-momentum equation is centered at the point $(i - 1/2, j)$, and the y-momentum equation is centered at $(i \cdot j - 1/2)$. The two equations are

$$\frac{u^{n+1}_{i-1/2,j} - u^{n}_{i-1/2,j}}{(\Delta t/2)} + \frac{(u^2)^{n+1}_{i,j} - (u^2)^{n+1}_{i-1,j}}{\Delta x} + \frac{(uv)^{n+1}_{i-1/2,j+1/2} - (uv)^{n+1}_{i-1/2,j-1/2}}{\Delta y}$$

$$= -\frac{p^{n+1}_{i,j} - p^{n+1}_{i-1,j}}{\Delta x} + \frac{1}{R}\left(\frac{u^{n+1}_{i+1/2,j} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-3/2,j}}{\Delta x^2}\right.$$

$$\left. + \frac{u^{n+1}_{i-1/2,j+1} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-1/2,j-1}}{\Delta y^2}\right) \qquad (2.6)$$

$$\frac{v^{n+1}_{i,j-1/2} - v^{n}_{i,j-1/2}}{(\Delta t/2)} + \frac{(uv)^{n+1}_{i+1/2,j-1/2} - (uv)^{n+1}_{i-1/2,j-1/2}}{\Delta x} + \frac{(v^2)^{n+1}_{i,j} - (v^2)^{n+1}_{i,j+1}}{\Delta y}$$

$$= -\frac{p^{n+1}_{i,j} - p^{n+1}_{i,j-1}}{\Delta y} + \frac{1}{R}\left(\frac{v^{n+1}_{i+1,j-1/2} - 2v^{n+1}_{i,j-1/2} + v^{n+1}_{i-1,j-1/2}}{\Delta x^2}\right.$$

$$\left. \frac{v^{n+1}_{i,j+1/2} - 2v^{n+1}_{i,j-1/2} + v^{n+1}_{i,j-3/2}}{\Delta y^2}\right) \qquad (2.7)$$

where $(n+1)$ is the new time station, (n) is the old time station, the factor of 1/2 in the time step is for convenience later, and the pressure for the moment is assumed to be known. After multiplying through by Δt we can define the following quantities.

$$C_x = \frac{\Delta t}{\Delta x}\ , \quad C_y = \frac{\Delta t}{\Delta y}\ , \quad D_x = \frac{\Delta t}{R\Delta x^2}\ , \quad D_y = \frac{\Delta t}{R\Delta y^2} \qquad (2.8)$$

Since the convective terms in eqs. (2.6) and (2.7) are nonlinear and we can only solve linear difference equations, we must linearize the equations. The linearized values of the dependent variables will be denoted by an overbar. Dealing with only eq. (2.6) for the present, we get

$$2\left(u^{n+1}_{i-1/2,j} - u^{n}_{i-1/2,j}\right) + C_x(\bar{u}_{i,j}u^{n+1}_{i,j} - \bar{u}_{i-1,j}u^{n+1}_{i-1,j})$$

$$+ C_y(\bar{v}_{i-1/2,j+1/2}u^{n+1}_{i-1/2,j+1/2} - \bar{v}_{i-1/2,j-1/2}u^{n+1}_{i-1/2,j-1/2})$$

$$= -C_x(p^{n+1}_{i,j} - p^{n+1}_{i-1,j}) + D_x(u^{n+1}_{i+1/2,j} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-3/2,j})$$

$$+ D_y(u^{n+1}_{i-1/2,j+1} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-1/2,j-1}) \qquad (2.9)$$

Since the values of u are only known at the half-integer values of i, and the values of v are known only at the integer values of j on the staggered grid, simple averages (which are formally second-order accurate) are used to eliminate these values in eq. (2.9),

$$2\left(u^{n+1}_{i-1/2,j} - u^{n}_{i-1/2,j}\right) + C_x[\bar{u}_{i,j} \cdot \tfrac{1}{2}(u^{n+1}_{i-1/2,j} + u^{n+1}_{i+1/2,j})$$

$$- \bar{u}_{i-1,j} \cdot \tfrac{1}{2}(u^{n+1}_{i-1/2,j} + u^{n+1}_{i-3/2,j})]$$

$$+ C_y[\bar{v}_{i-1/2,j+1/2} \cdot \tfrac{1}{2}(u^{n+1}_{i-1/2,j+1} + u^{n+1}_{i-1/2,j})$$

$$- v_{i-1/2,j-1/2} \cdot \tfrac{1}{2}(u^{n+1}_{i-1/2,j} + u^{n+1}_{i-1/2,j-1})] = -C_x(p^{n+1}_{i,j} - p^{n+1}_{i-1,j})$$

$$+ D_x(u^{n+1}_{i+1/2,j} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-3/2,j})$$

$$+ D_y(u^{n+1}_{i-1/2,j+1} - 2u^{n+1}_{i-1/2,j} + u^{n+1}_{i-1/2,j-1}) \qquad (2.10)$$

where the averages are used in calculating the linearized (barred) quantities also.

After regrouping the terms in eq. (2.10), the final form of this difference equation can be written

$$A_N^U u_{i-1/2,j+1}^{n+1} + A_W^U u_{i-3/2,j}^{n+1} + (2 + A_{PX}^U + A_{PY}^U) u_{i-1/2,j}^{n+1} + A_E^U u_{i+1/2,j}^{n+1}$$

$$+ A_S^U u_{i-1/2,j-1}^{n+1} - u_{i-1/2,j}^{n} + C_x(p_{i,j}^{n+1} - p_{i-1,j}^{n+1}) = 0 \qquad (2.11)$$

where

$$A_N^U = \frac{1}{2} C_y \bar{v}_{i-1/2,j+1/2} - D_y; \qquad A_S^U = -\frac{1}{2} C_y \bar{v}_{i-1/2,j-1/2} - D_y$$

$$A_W^U = -\frac{1}{2} C_x \bar{u}_{i-1,j} - D_x; \qquad A_E^U = \frac{1}{2} C_x \bar{u}_{i,j} - D_x \qquad (2.12)$$

$$A_{PX}^U = \frac{1}{2} C_x(\bar{u}_{i,j} - \bar{u}_{i-1,j}) + 2D_x; \quad A_{PY}^U = \frac{1}{2} C_y(\bar{v}_{i-1/2,j+1/2} - \bar{v}_{i-1/2,j-1/2}) + 2D_y$$

Similarly, the final form of the v-momentum equation, (2.7), can be written

$$A_N^V v_{i,j+1/2}^{n+1} + A_W^V v_{i-1,j-1/2}^{n+1} + (2 + A_{PX}^V + A_{PY}^V) v_{i,j-1/2}^{n+1} + A_E^V v_{i+1,j-1/2}^{n+1}$$

$$+ A_S^V v_{i,j-3/2}^{n+1} - v_{i,j-1/2}^{n} + C_y(p_{i,j}^{n+1} - p_{i,j-1}^{n+1}) = 0 \qquad (2.13)$$

where

$$A_N^V = \frac{1}{2} C_y \bar{v}_{i,j} - D_y; \qquad A_S^V = -\frac{1}{2} C_y \bar{v}_{i,j-1} - D_y$$

$$A_W^V = -\frac{1}{2} C_x \bar{u}_{i-1/2,j-1/2} - D_x; \qquad A_E^V = \frac{1}{2} C_x \bar{u}_{i+1/2,j-1/2} - D_x \qquad (2.14)$$

$$A_{PX}^V = \frac{1}{2} C_x(\bar{u}_{i+1/2,j-1/2} - \bar{u}_{i-1/2,j-1/2}) + 2D_x;$$

$$A_{PY}^V = \frac{1}{2} C_y(\bar{v}_{i,j} - \bar{v}_{i,j-1}) + 2D_y$$

The only remaining equation is the continuity equation, (2.3). This is placed in finite-difference form by centering the equation at the point (i,j).

$$u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1} + \frac{\Delta x}{\Delta y}(v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}) = 0 \qquad (2.15)$$

These three equations, (2.11), (2.13) and (2.13) form the basis for the pressure corrector method.

The implementation of the method follows from the following argument. If the correct pressure were known, then eqs. (2.11) and (2.13) could be solved for u and v, and eq. (2.15) would be satisfied exactly. However, it is not known, so we have to make a guess for it, and then correct it in a rational way to satisfy eq. (2.15) which we want to hold. Hence, we assume that the true solution is given by

$$\begin{aligned} u^{n+1} &= u^* + u' \\ v^{n+1} &= v^* + v' \\ p^{n+1} &= p^* + p' \end{aligned} \tag{2.16}$$

where the starred quantities are the guesses and the primed quantities are the corrections. Placing (2.16) into eq. (2.11) for u yields

$$\begin{aligned} &A_N^U(u^* + u')_{i-1/2,j+1} + A_W^U(u^* + u')_{i-3/2,j} + (\tilde{2} + A_{PX}^U + A_{PY}^U)(u^* + u')_{i-1/2,j} \\ &+ A_E^U(u^* + u')_{i+1/2,j} + A_S^U(u^* + u')_{i-1/2,j-1} - u^n_{i-1/2,j} \\ &\qquad + C_x[(p^* + p')_{i,j} - (p^* + p')_{i-1,j}] = 0 \end{aligned} \tag{2.17}$$

Hence, the equation for the predicted (guessed) value of u^* is identical to equation (2.11) but with $(n+1)$ values replaced by starred values, since eq. (2.17) is linear and can be split into a predictor equation and a corrector equation. The remaining terms in eq. (2.17) give the full equation for the correction value u':

$$A^U_N u'_{i-1/2,j+1} + A^U_W u'_{i-3/2,j} + (2 + A^U_{PX} + A^U_{PY}) u'_{i-1/2,j}$$

$$+ A^U_E u'_{i+1/2,j} + A^U_S u'_{i-1/2,j-1} + C_x(p'_{i,j} - p'_{i-1,j}) = 0 \qquad (2.18)$$

This equation, as it stands, is of no value, but an approximate form of the equation leads to the final solution. For more details of the following argument, see Chorin[3] or Amsden and Harlow[4]. In order to alter the velocity, but not change the value of the vorticity, the velocity can only be corrected by the gradient of a scalar function. Hence, eq. (2.18) is approximated by

$$u'_{i-1/2,j} = -\frac{C_x}{(2 + A^U_{PX} + A^U_{PY})}(p'_{i,j} - p'_{i-1,j}) \qquad (2.19)$$

In an analogous way for the v-momentum equation, it can be shown that the predicted value of v^* is obtained from eq. (2.13) with $(n+1)$ replaced by starred values, and the correction for v from

$$v'_{i,j-1/2} = -\frac{C_y}{(2 + A^V_{PX} + A^V_{PY})}(p'_{i,j} - p'_{i,j-1}) \qquad (2.20)$$

Note in passing here that had the values for a predicted and corrected value been placed in either the full differential equations (2.4) and (2.5) or the nonlinear difference equations (2.6) and (2.7) the results for the correction values would have been significantly more complicated.

To determine the equation for the correction (primed values), we must account for the fact that the continuity equation has not yet been satisfied. Thus, placing eqs. (2.16) for u^{n+1} and v^{u+1} into the differenced continuity equation, (2.15), and using the relations (2.19) and (2.20) for the correction velocity we get

$$-\frac{\Delta x}{\Delta y} B_N^Y p'_{i,j+1} - B_P^X p'_{i-1,j} + [B_E^X + B_P^X + \frac{\Delta x}{\Delta y}(B_N^Y + B_P^Y)]p'_{i,j} - B_E^X p'_{i+1,j}$$

$$-\frac{\Delta x}{\Delta y} B_P^Y p'_{i,j-1} + \Delta^* = 0 \qquad (2.21)$$

where

$$B_P^X = \left[\frac{C_x}{(2 + A_{PX}^U + A_{PY}^U)}\right]_P \qquad B_E^X = \left[\frac{C_x}{(2 + A_{PX}^U + A_{PY}^U)}\right]_E$$

$$B_P^Y = \left[\frac{C_y}{(2 + A_{PX}^V + A_{PY}^V)}\right]_P \qquad B_N^Y = \left[\frac{C_y}{(2 + A_{PX}^V + A_{PY}^V)}\right]_N \qquad (2.22)$$

$$\Delta^* = u^*_{i+1/2,j} - u^*_{i-1/2,j} + \frac{\Delta x}{\Delta y}(v^*_{i,j+1/2} - v^*_{i,j-1/2})$$

Clearly (2.21) is a difference form of a Poisson equation for the pressure.

2.3 Solution Algorithm

The solution procedure for this predictor-corrector scheme is as follows: (a) Guess a value of the pressure, e.g. the pressure at the last integration station; (b) compute the linearized quantities present; also from last integration station; (c) integrate the two momentum equations, subject to the imposed boundary conditions discussed below in section 2.4, to get the values of u^* and v^*; (d) find p' from the Poisson equation, (2.21); (e) correct the starred quantities using eq. (2.16) to get the new values of u, v, p; (f) continue this process until a satisfactory convergence is obtained.

Step (c) above requires the solution of the two momentum equations which now have the form

$$A_N^U u^*_{i-1/2,j+1} + A_W^U u^*_{i-3/2,j} + (2 + A_{PX}^U + A_{PY}^U)u^*_{i-1/2,j} + A_E^U u^*_{i+1/2,j}$$

$$+ A_S^U u^*_{i-1/2,j-1} - u^n_{i-1/2,j} + C_x(p^*_{i,j} - p^*_{i-1,j}) = 0 \qquad (2.23)$$

$$A^V_N v^*_{i,j+1/2} + A^V_W v^*_{i-1,j-1/2} + (2 + A^V_{PX} + A^V_{PY}) v^*_{i,j-1/2} + A^V_E v^*_{i+1,j-1/2}$$

$$+ A^V_S v^*_{i,j-3/2} - v^n_{i,j-1/2} + C_y(p^*_{i,j} - p^*_{i,j-1}) = 0 \qquad (2.24)$$

Rather than solving these in this fully implicit form, in which they were written for convenience in deriving the predictor and corrector steps of the methods, an alternating direction procedure will be used to generate a scheme where only tridiagonal matrices need to be solved for the unknown values. Writing eqs. (2.23) and (2.24) in a more convenient shorthand notation

$$L_x q_{i-1,j} + L_y q_{i,j-1} + (2 + M_x + M_y) q_{i,j} + N_x q_{i+1,j} + N_y q_{i,j+1} - q^n_{i,j}$$

$$+ D = 0 \qquad (2.25)$$

where the correspondence between (2.25) and either (2.23) or (2.24) can be found by inspection, and only the relative (i,j) positions of the unknowns on the difference mesh of figure 1 has been shown. The factor 2, in the middle term above, which was introduced into the time differencing of eqns. (2.6) and (2.7) will now be seen to allow a symmetric ADI scheme to be formed. Eq. (2.25) can be split into the following ADI scheme

$$(1 + F_x)\bar{q} + (-1 + F_y) q^n + D = 0 \qquad (2.26a)$$

$$(1 + F_y) + (-1 + F_x)\bar{q} + D = 0 \qquad (2.26b)$$

which is equivalent, to second-order accuracy, to a Crank-Nicolson scheme expressed as

$$q - q^n + (F_x + F_y)(q + q^n) + 2D = 0 \qquad (2.26c)$$

where

$$F_x = L_x q_{i-1} + M_x q_i + N_x q_{i+1} \tag{2.27a}$$

$$F_y = L_y q_{j-1} + M_y q_j + N_y q_{j+1} \tag{2.27b}$$

Thus, equations (2.26) can be looked upon as a Peaceman-Rachford[10] ADI solution procedure for either u^* or v^*, and this only requires the use of a two-sweep, or Thomas, algorithm for the inversion of tridiagonal matrices.

The Poisson equation for the pressure correction, eq. (2.21), needs to be solved in step (d) above. In the time asymptotic solution of steady-state problems, it has been found that the Poisson equation need not be solved exactly at each fictitious time step, but that only a reasonably consistent value of p' need be generated by the Poisson solver. This is accomplished by turning the elliptic Poisson equation into a nonhomogeneous, parabolic equation for p', and advancing this equation a few (usually 3-5) time steps for the value of p'. Hence the Poisson equation, (2.21), can be immediately cast into a form exactly equivalent to eq. (2.26) by adding $2/\Delta\tau\ p'_{i,j}$ and subtracting $2/\Delta\tau\ p^n_{i,j}$ to equation (2.21). Now the ADI formalism can be used to march the converted eq. (2.21) three to five steps for a consistent p' and the solution procedure outlined at the beginning of this section can proceed.

2.4 Boundary Conditions

Although the use of the primitive variable formulation allows the direct specification of the velocity values on the boundaries, rather than indirectly as in the case of the vorticity and stream-function formulation, the use of the staggered grid depicted in figure 1 complicates things. Were all velocity points coincident with the boundary, specification would be trivial. However

since half the velocity nodes needed to specify boundary conditions lie outside the computational domain, special relations are necessary to provide the correct boundary values at the midpoint between the velocity nodes. On the other hand, the boundary conditions on the pressure (actually the pressure correction, as will be seen) are quite simple, and easily applied.

In this section, the four boundaries of the domain shown in figure 1 are denoted by the obvious terms upper, lower, left and right boundaries. Conditions for u on each of these will be described first, followed by conditions for v on each. Next will be a discussion of the pressure boundary conditions, and the consequences on the pressure of using the pressure corrector method. Finally, the incorporation of typical boundary conditions into the implicit solution algorithm described in section 2.3 will be shown.

The u-velocity nodes fall directly on the boundaries on the left and right sides of the domain. Thus, if the velocity is given on these boundaries, for example a solid wall, or a specified inflow profile, the value is fixed at the given value. If outflow is specified at the right boundary, i.e. $\partial u/\partial x = 0$, then the following relations are used to connect the variation of the boundary velocity with two internal points.

$$u_{IMAX+1,j} = (4u_{IMAX,j} - u_{IMAX-1,j})/3 \qquad (2.28)$$

This relation is second-order accurate and does not destroy the tridiagonal system of equations when implemented as shown below.

For all problems discussed here, the upper and lower boundaries can be considered solid walls. Thus the u velocity on these walls is given by the wall velocity due to the no-slip condition. Unfortunately, the boundary

is located between u nodes, see figure 1, and hence the u-value given there cannot be directly specified. Thus, a relation giving u at the boundary in terms of the exterior and interior points must be derived. The standard relation which has been recommended for this problem is based on a simple average. For the top wall, denoting the velocity at the wall by subscript w, we can write

$$u_w = \frac{1}{2}\left(u_{i,JMAX+1} + u_{i,JMAX}\right)$$

Hence the boundary condition to be used for the unknown velocity at node JMAX+1 can be expressed as

$$u_{i,JMAX+1} = 2u_w - u_{i,JMAX} \tag{2.29}$$

This is a formally second-order accurate result. However, a higher-order relation which still retains the tridiagonal form can be derived. This gives

$$u_{i,JMAX+1} = \frac{8}{3}u_w - 2u_{i,JMAX} + \frac{1}{3}u_{i,JMAX-1} \tag{2.30}$$

Now, although the recommended relation, (2.29) is formally second-order accurate, consider the case of a fully developed pipe (Poiseuille) flow. Here the u velocity has a parabolic form. Equations (2.29) implies that the velocity variation around the wall point is linear, which in the case of Poiseuille flow, it certainly is not. Relation (2.30) was derived by allowing a quadratic variation around the wall and is consistent with a parabolic profile, such as Poiseuille flow. The use of these on the test problems will be discussed in section 2.5 where the results are presented.

The situation for v boundary conditions is very similar to what was just stated for u above, except with i and j reversed. The upper and lower walls conicide with v nodes and so v can be specified here exactly. For all cases considered here $v = 0$ on both walls. The left boundary is

either a wall or a specified inflow, and so v_w is known. Thus an appropriate version of eqs. (2.29) or (2.30) for v at $i = 1$ can be used. The right boundary, if it is a wall, uses eq. (2.29) or eq. (2.30) with u replaced by v and i and j unchanged. If it is an outflow boundary, then $\partial v/\partial x = 0$ must be used in order to specify $v_{IMAX+1,j}$. Here, the standard relation given is the only one which could be found to satisfy this gradient condition, consistent with maintaining the tridiagonal form of the system, i.e.,

$$v_{IMAX+1,j} = v_{IMAX,j} \tag{2.31}$$

The boundary conditions necessary for the solution of the pressure corrector equation, (2.21) follow directly from the definitions of the velocity corrections, eqs. (2.19) and (2.20). Since in all cases here the velocity is to be specified on every boundary, we solve the equations for the predicted (starred) velocities by setting u^* and v^* equal to the correct boundary conditions. Hence, the correction (primed) velocities are all zero on the boundaries. However, using eqs. (2.19) and (2.20) these correction velocities are related to the gradient of the correction pressure, so at the left and right boundaries where u is specified, we have $\partial p'/\partial x = 0$, and on the upper and lower boundaries $\partial p'/\partial y = 0$. Thus the Poisson equation for the pressure correction is solved subject to Neumann boundary conditions.

Note that these conditions are on the pressure correction, p', and not the guessed p^* or corrected pressure p. In fact, in the formulation used and described in this report, the boundary values of p or p^* are never used. Checking the differencing and the grid will confirm this. This fact, which does not seem to have been emphasized before, is a plausible result of solving the equations in primitive variables. Most pressure boundary conditions are derived from the equations of motion since the actual pressure on a boundary is not known a priori in any flow situation. They are considered

"extra" conditions necessary to complete the particular mathematical (numerical) formulation being used. In fact, they are extra, because in the vorticity stream-function formulation a fourth-order biharmonic equation for the stream function can be generated. This requires only two boundary conditions at each boundary surface. They are the function and normal derivative, which are in fact u and v. No further conditions are necessary. Essentially the same thing is done in the "pressure corrector" method. The velocities are given as conditions, and then a consistent set of boundary conditions on a non-physical variable is derived and used to compute the relevant physical quantity, i.e., the pressure. Never is any recourse needed for a "pressure" boundary condition. In this way, the pressure corrector scheme models the physics of the flow closely in that an internally generated value of the pressure is obtained at all points inside the solution domain without any reference to boundary values of pressure. If the value of the pressure on the boundary (which lies between pressure nodes) is desired, an extrapolation from interior points can be used.

To incorporate any of these boundary conditions into the solution algorithm the following method is used, which keeps the tridiagonal form of the matrix intact, and allows the use of the tridiagonal solver given in the Appendix. Using the notation of the Appendix, the general form of the difference equations generated in this section are

$$A_K Q_{K-1} + B_K Q_K + C_K Q_{K+1} + D_K = 0 \qquad (2.32)$$

where the Q are the unknown variables and the A, B, C, D are coefficients computed at the K location. It is presumed that boundary values for Q (either computed or prescribed) are given at a starting location, KST, and an ending location, KND. The tridiagonal solver is then invoked from KST+1 to KND-1. Thus, at the beginning and final stations of the solver's use,

eq. (2.32), appears

$$A_{KST+1}Q_{KST} + B_{KST+1}Q_{KST+1} + C_{KST+1}Q_{KST+2} + D_{KST+1} = 0 \qquad (2.33a)$$

$$A_{KND-1}Q_{KND-2} + B_{KND-1}Q_{KND-1} + C_{KND-1}Q_{KND} + D_{KND-1} = 0 \qquad (2.33b)$$

However, at each of these stations, the boundary value is either known outright, or is given in terms of the two next interior points. As an example writing boundary condition (2.30) in general form consistent with (2.32) and (2.33) gives

$$Q_{KND} = bQ_{KND-1} + aQ_{KND-2} + d \qquad (2.34)$$

This is in fact, the most general boundary condition which can be incorporated, including all conditions from fixed values, for which $b = c = 0$, to forms like eq. (2.30). Placing eq. (2.34) into (2.33b) gives the following equation at the last point where the solver is employed.

$$\bar{A}_{KND-1}Q_{KND-2} + \bar{B}_{KND-1}Q_{KND-1} + \bar{D}_{KND-1} = 0 \qquad (2.35a)$$

where

$$\bar{A}_{KND-1} = A_{KND-1} + aC_{KND-1} \qquad (2.35b)$$

$$\bar{B}_{KND-1} = B_{KND-1} + bC_{KND-1} \qquad (2.35c)$$

$$\bar{D}_{KND-1} = D_{KND-1} + dC_{KND-1} \qquad (2.35d)$$

Thus eq. (2.35a) is in the correct form for the solver to proceed, and there is a similar equation at KST+1. All that is required is that the coefficients A, B, C, D be modified to incorporate the boundary values as in (2.35b) - (2.35c) at the two end points KST+1 and KND-1, and after the solver generates the values of Q_K $(KST+1) \leq K \leq (KND-1)$, a separate calculation generates Q_{KST} and Q_{KND} from the definitions of the boundary values, such as eq. (2.34).

3.0 VORTICITY STREAM FUNCTION FORMULATION

3.1 Equations and Grid

The equations used to solve for the vorticity and stream function as dependent variables are easily derived from the incompressible primitive variable form of the Navier Stokes equations. The vorticity, for the two dimensional case, is defined by the single component

$$\zeta = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \tag{3.1}$$

An equation for this variable can be obtained by cross-differentiating equations (2.1) and (2.2) and eliminating the pressure to yield

$$\frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} = \frac{1}{R}\left(\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2}\right) \tag{3.2}$$

Again, the time derivative is included in the equations as a means of reaching the asymptotic steady state which is desired; no true time dependent solutions are sought. Using the definition of the stream function in two dimensions

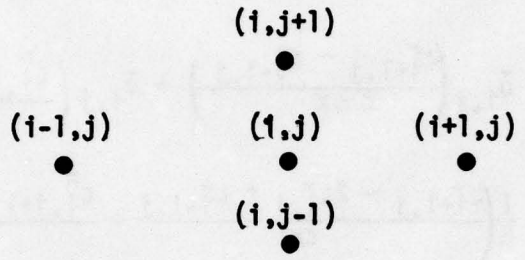
$$u = \frac{\partial \psi}{\partial y} \quad , \qquad v = -\frac{\partial \psi}{\partial x} \tag{3.3}$$

the continuity equation, eqn (2.3) is automatically satisfied, and the equation which determines ψ can be found by placing eqn (3.3) into definition (3.1) to yield

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \zeta \tag{3.4}$$

These two equations (3.2) and (3.4), along with the auxiliary relations (3.3) for the velocity, are the equations to be used in this formulation of the incompressible Navier-Stokes equations.

The finite-difference form of these equations is solved on a standard, five point, mesh, depicted below



In contrast to the primitive variable procedure, here all the dependent variables are located at the same points, and the boundaries of the solution domain pass through the first and last row or column of nodes. Thus the boundary values are imposed at the boundary, and no interpolation is required.

However, the boundary conditions for the vorticity stream function formulation still are not applicable in a straightforward manner. We need conditions on all boundaries for ψ and ζ. What we have are no slip conditions for u and v. The normal velocity condition can be translated into a condition for ψ, but the vorticity boundary condition is still lacking. The method for determining the boundary vorticity is described in Section 3.4 below, and is one of the standard means of accomplishing this.

3.2 Finite Difference Equations

Both the 'time dependent' equation for vorticity transport, (3.2), and the Poisson equation for the stream function (3.4) are solved by using central differences in space. Unlike the primitive variable solution

procedure, these equations are solved directly for the dependent field variables, and so the time integration in eq. (3.2) can be done immediately by an ADI procedure. Since the convection turns on the left hand side are non-linear they are handled in much the same way as in the primitive variable section, where an overbar was used to denote a dependent variable which has been linearized. Thus, the difference equations appear:

$$\frac{\zeta^*_{i,j} - \zeta^n_{i,j}}{\Delta t/2} + \bar{u}_{i,j}\left(\frac{\zeta^*_{i+1,j} - \zeta^*_{i-1,j}}{2\,\Delta x}\right) + \bar{v}_{i,j}\left(\frac{\zeta^n_{i,j+1} - \zeta^n_{i,j-1}}{2\,\Delta y}\right) = \frac{1}{R}\left(\frac{\zeta^*_{i+1,j} - 2\zeta^*_{i,j} + \zeta^*_{i-1,j}}{\Delta x^2} + \frac{\zeta^n_{i,j+1} - 2\zeta^n_{i,j} + \zeta^n_{i,j-1}}{\Delta y^2}\right) \tag{3.5a}$$

$$\frac{\zeta^{n+1}_{i,j} - \zeta^*_{i,j}}{\Delta t/2} + \bar{u}_{i,j}\left(\frac{\zeta^*_{i+1,j} - \zeta^*_{i-1,j}}{2\Delta x}\right) + \bar{v}_{i,j}\left(\frac{\zeta^{n+1}_{i,j+1} - \zeta^{n+1}_{i,j-1}}{2\Delta y}\right) = \frac{1}{R}\left(\frac{\zeta^*_{i+1,j} - 2\zeta^*_{i,j} + \zeta^*_{i-1,j}}{\Delta x^2} + \frac{\zeta^{n+1}_{i,j+1} - 2\zeta^{n+1}_{i,j} + \zeta^{n+1}_{i,j-1}}{\Delta y^2}\right) \tag{3.5b}$$

These equations are already in the tridiagonal form, discussed in section 2.3, which allows for their rapid solution. This can be displayed more directly by combining terms in (3.5) above to yield

$$L_x\,\zeta^*_{i-1,j} + M_x\,\zeta^*_{i,j} + N_x\,\zeta^*_{i+1,j} + R_x = 0 \tag{3.6a}$$

$$L_y\,\zeta^{n+1}_{i,j-1} + M_y\,\zeta^{n+1}_{i,j} + N_y\,\zeta^{n+1}_{i,j+1} + R_y = 0 \tag{3.6b}$$

where

$$L_x = -\frac{1}{4} C_x \bar{u}_{i,j} - \frac{1}{2} D_x; \quad M_x = 1 + D_x; \quad N_x = \frac{1}{4} C_x \bar{u}_{i,j} - \frac{1}{2} D_x \qquad (3.7a)$$

$$R_x = -\zeta^n_{i,j} + \frac{1}{4} C_y \bar{v}_{i,j} (\zeta^n_{i,j+1} - \zeta^n_{i,j-1}) - \frac{1}{2} D_y (\zeta^n_{i,j+1} - 2\zeta^n_{i,j} + \zeta^n_{i,j-1})$$

$$L_y = -\frac{1}{4} C_y \bar{v}_{i,j} - \frac{1}{2} D_x; \quad M_y = 1 + D_y; \quad N_y = \frac{1}{4} C_y \bar{v}_{i,j} - \frac{1}{2} D_y \qquad (3.7b)$$

$$R_y = -\zeta^*_{i,j} + \frac{1}{4} C_x \bar{u}_{i,j} (\zeta^*_{i+1,j} - \zeta^*_{i-1,j}) - \frac{1}{2} D_x (\zeta^*_{i+1,j} - 2\zeta^*_{i,j} + \zeta^*_{i-1,j})$$

and

$$C_x = \frac{\Delta t}{\Delta x}; \quad C_y = \frac{\Delta t}{\Delta y}; \quad D_x = \frac{\Delta t}{R\Delta x^2}; \quad D_y = \frac{\Delta t}{R\Delta y^2} \qquad (3.8)$$

The linearized values of u and v are determined from the solution of the Poisson equation for the stream function, using the auxiliary relations (3.3) which define u and v. As in the primitive variable procedure, only the steady solution is desired, so the Poisson equation need not be solved exactly at each step. Only a solution for ψ, reasonably consistent with the current solution for ζ, is required for the method to proceed. Hence there is no time advantage to an implicit solution procedure for the solution of the Poisson equation since the exact solution (which could be obtained more quickly via the use of implicit methods) is not required. For the present purposes either an SOR or explicit integration technique was used. Both can be written as

$$\psi^{n+1}_{i,j} = \psi^n_{i,j} + \frac{\Omega}{4}\Big[\psi^n_{i+1,j} - 2\psi^n_{i,j} + \psi^n_{i-1,j} + \frac{\Delta x^2}{\Delta y^2}(\psi^n_{i,j+1} - 2\psi^n_{i,j} + \psi^n_{i,j-1}) - \Delta x^2 \psi^{n+}_{i,j}\Big] \qquad (3.9)$$

where Ω is the relaxation factor for SOR or equivalent to $4\Delta\tau/\Delta x^2$ if an explicit scheme is used, where $\Delta\tau$ is a false time.

3.3 Solution Algorithm

The solution procedure using the ADI-SOR combination just described is as follows: (a) Compute the linearized velocities, u and v, from the last computed values of ψ; (b) Integrate the vorticity equation one complete ADI step, subject to the imposed boundary conditions discussed below in Section 4, using the standard tridiagonal solver, at each of the two intermediate steps which comprise a complete ADI step; (c) Find a new value of ψ, consistent with the value of ζ found in step (b), by making three to five SOR passes (or explicit time steps) through the Poisson equation for ψ; (d) Continue this process until a satisfactory convergence is obtained.

3.4 Boundary Conditions

The difficulty with the $\psi - \zeta$ formulation of the Navier Stokes equations is that the natural boundary conditions we have are on u and v, not ψ or ζ. Both of these velocity conditions can be easily related to conditions on ψ, but neither one directly relates to ζ. Thus, the boundary conditions on the velocity normal to the boundaries leads directly to a specification of ψ at all points on the boundary. However, unless we know the vorticity exactly, as in the case of a specified velocity profile at an inflow boundary; or can specify its normal gradient equal to zero, as a result of a downstream continuation outflow boundary, there is no analytic specification of ζ at a boundary which can be made without recourse to the equations themselves and other boundary conditions. But, to complete the specification of the Navier-Stokes problem to be solved, a condition for the ζ variable on the boundary (either ζ or its normal derivative given) must be specified.

For inflow boundaries, as previously stated, given the velocity profile, it can easily be integrated tangential to the boundary to get ψ, and differentiated to specify ζ. At outflow boundaries, the standard, second order accurate, one sided, first derivative form can be used to specify both $\partial\psi/\partial n$ and $\partial\zeta/\partial n$ equal to zero, where n denotes the normal to the boundary. Taking for example, x to be the direction of outflow, the following relation for ζ (or ψ) can be obtained which does not destroy the tridiagonal form of the matrix to be solved.

$$\zeta_{IMAX_j} = \frac{4}{3}\zeta_{IMAX-1,j} - \frac{1}{3}\zeta_{IMAX-2,j} \qquad (3.10)$$

This is in the general form used in (2.34) to incorporate boundary conditions into the tridiagonal solver, and the implementation is the same as described following eqn (2.34) in section 2.4.

For solid walls, we can still specify ψ by knowing that the walls are streamlines and if there is any mass injection (normal velocity) through the walls. However, since we do not know the normal derivative of the tangential velocity (although we do know the velocity) the vorticity is not known directly. A condition for ζ_w must be made which is consistent with the known boundary conditions and the governing equations. This can be done in a variety of ways, see Roache [11]; here we use the most straightforward. The unused condition on the tangential velocity is combined with the governing differential equation, written in difference form on the boundary in question, to generate a relation for the wall vorticity, ζ_w as follows. The Poisson equation for ψ, eqn (3.4), can be written at any boundary, in normal and tangential coordinates as,

$$\frac{\partial^2\psi_w}{\partial n^2} + \frac{\partial^2\psi_w}{\partial t^2} = \zeta_w$$

In most cases (all that are considered here), ψ_w is a specified constant, so that its tangential derivative along the wall is zero. Writing the remaining second derivative in difference form, centered <u>at the wall</u> gives

$$\psi_+ - 2\psi_w + \psi_- = (\Delta n)^2 \zeta_w \tag{3.11}$$

in obvious notation. The value of either ψ_+ or ψ_- is not known, depending upon the location of the boundary, i.e. the end or beginning of the domain. However, we do have a relation which involves this external point, namely the stream function definition of the tangential velocity, q_t. In difference form this is

$$q_{t_w} = \frac{\psi_+ - \psi_-}{2\Delta n} \tag{3.12}$$

Solving eqn (3.12) for either ψ_+ or ψ_- and placing it into eqn (3.11) yields the boundary relation for ζ_w, either

$$\zeta_w = \frac{2}{\Delta n}\left[q_{t_w} - \frac{\psi_w - \psi_-}{\Delta n}\right] \tag{3.13a}$$

at the end of the domain, or

$$\zeta_w = \frac{2}{\Delta n}\left[\frac{\psi_+ - \psi_w}{\Delta n} \quad q_{t_w}\right] \tag{3.13b}$$

at the beginning of the domain.

These are only first order accurate conditions for ζ_w but have been shown to be sufficiently accurate in most cases. These relations for ζ are decoupled from the current solution for ψ and lead to the sequential solution algorithm given in Section 2.3.

4.0 BIHARMONIC FORMULATION

4.1 Equations and the Newton-Chord Method

Eliminating the vorticity ζ in the vorticity-stream function formulation yields what we call the biharmonic formulation:

$$\text{a.} \qquad \frac{\partial}{\partial t} \nabla^2 \psi = \mathcal{N}(\psi) \tag{4.1}$$

where

$$\text{b.} \qquad \mathcal{N}(\psi) \equiv \frac{1}{R} \nabla^4 \psi - \frac{\partial \psi}{\partial y} \nabla^2 \frac{\partial \psi}{\partial x} + \frac{\partial \psi}{\partial x} \nabla^2 \frac{\partial \psi}{\partial y}$$

For the steady state we have simply

$$\mathcal{N}(\psi) = 0 \tag{4.2}$$

Newton's method for solving such a nonlinear problem assumes given a current approximation to the solution, say $\psi = F$, and then we seek a correction, say ϕ, such that $\psi = F + \phi$ satisfies (4.2) when second and higher-order terms in ϕ are dropped. This procedure gives the linear problem for ϕ:

$$\mathcal{L}(F)\phi = -\mathcal{N}(F) \tag{4.3}$$

where the linear operator $L(F)$ is defined by:

$$\mathcal{L}(F) \equiv \frac{1}{R} \nabla^4 + \left(\nabla^2 \frac{\partial F}{\partial y} - \frac{\partial F}{\partial y} \nabla^2\right) \frac{\partial}{\partial x} - \left(\nabla^2 \frac{\partial F}{\partial x} - \frac{\partial F}{\partial x} \nabla^2\right) \frac{\partial}{\partial y} \tag{4.4}$$

Of course the indicated procedure can be iterated (replacing F by $F + \phi$, which is Newton's method) and terminated when, say $\| \phi \| < 10^{-d}/2$ if d digits are required. This method has many virtues; not the least of which is rapid convergence. Specifically the error at the next application is essentially $\| \phi \|^2$ if F is sufficiently near a solution.

The main difficulty with Newton's method is that at each iterate a new linear problem (4.3) - (4.4) must be solved. However, the so-called modified

Newton method or Chord method can be used to circumvent this. Specifically, the linear operator is "frozen" at say $\mathcal{L}(F_0) \equiv \mathcal{L}_0$, the first iterate, and we solve in place of (4.3) the system

$$\mathcal{L}_0\phi = -\mathcal{N}(F) \tag{4.5}$$

Now only the right-hand side need be updated at each iteration and the main work in solving (4.5) need be done only once. It is easy to show that this procedure, although not quadratically convergent, does converge quite fast if F_0 is reasonably near a solution. Techniques have been developed to insure that this combined Newton-Chord iteration, or slight modifications of it do converge and that no more than two full Newton iterates need be done!

In the formulation and discussion above, we have neglected the boundary conditions which must be imposed along with (4.3) or (4.5). However, we will consider only linear boundary conditions, which will usually be ψ and its normal derivative prescribed (but may also involve the second and perhaps higher normal derivatives). Then it is always possible to arrange that each iterate satisfies the boundary conditions exactly, so that the correction, ϕ, merely has to satisfy the corresponding homogeneous boundary conditions.

4.2 Difference Approximations

We cover the relevant rectangular domain in the x,y plane by a grid system defined by

$$x_i = x_1 + (i-1)h \quad (i = 1, \ldots, m), \quad y_j = y_1 + (j-1)k \quad (j = 1, \ldots, n) \tag{4.6}$$

where h,k are the grid sizes in the x,y directions (see Fig. 2). We use standard second-order centered difference approximations. Thus it is convenient to introduce the central difference operators δ_h^2, δ_k^2, $\bar{\delta}_h$, $\bar{\delta}_k$ and

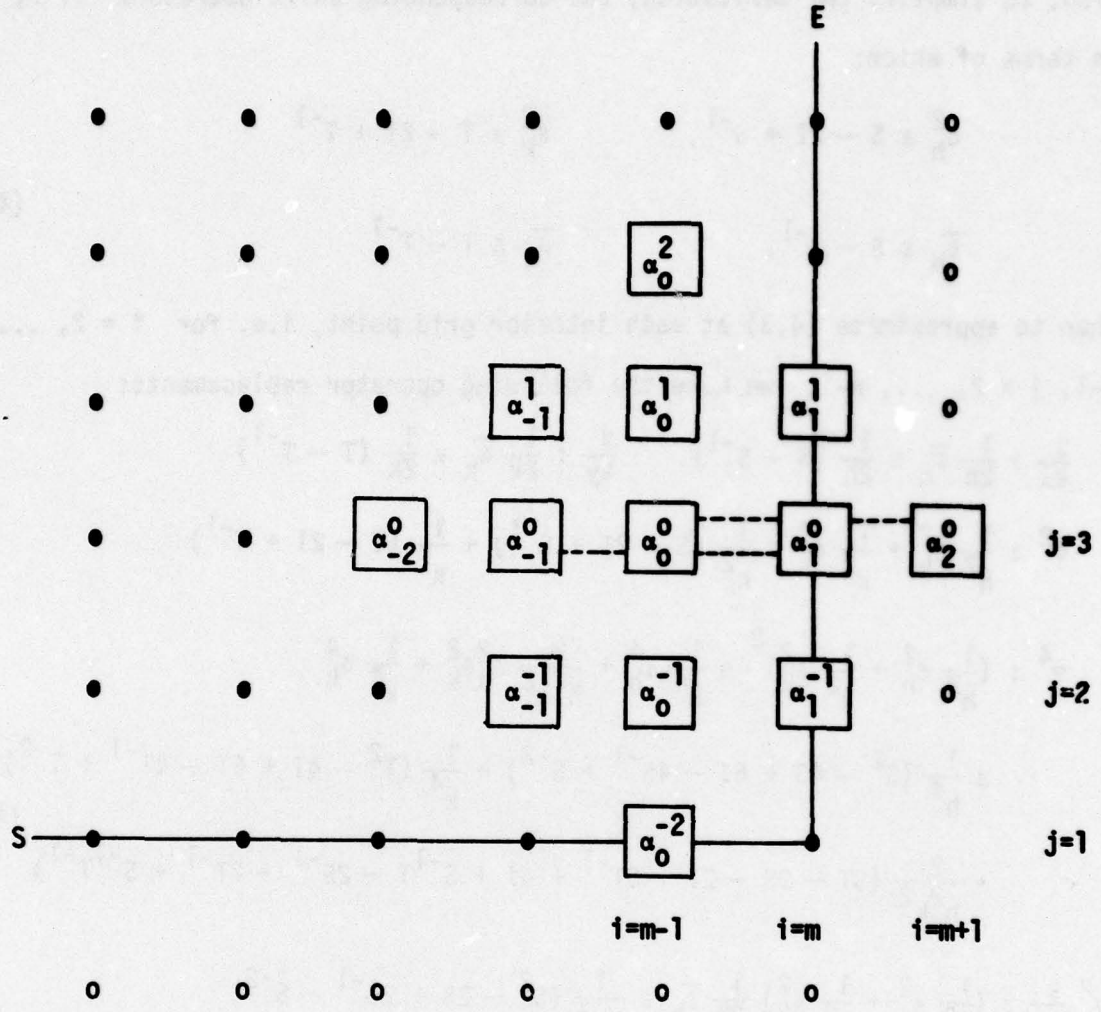


Figure 2. A typical computational molecule at a point where an exterior unstored grid value is used. The dotted lines indicate the grid values connected by the two boundary equations (4.27). Stored and unstored grid values in the calculation are indicated by • and o.

also, to simplify the derivation, the corresponding shift operators S, T, in terms of which:

$$\delta_h^2 \equiv S - 2I + S^{-1}, \qquad \delta_k^2 \equiv T - 2I + T^{-1}$$
$$\bar{\delta}_h \equiv S - S^{-1}, \qquad \bar{\delta}_k \equiv T - T^{-1} \tag{4.7}$$

Then to approximate (4.3) at each interior grid point, i.e. for $i = 2, \ldots, m-1$, $j = 2, \ldots, n-1$, we make the following operator replacements:

$$\frac{\partial}{\partial x} : \frac{1}{2h}\bar{\delta}_h \equiv \frac{1}{2h}(S - S^{-1}), \qquad \frac{\partial}{\partial y} : \frac{1}{2k}\bar{\delta}_k \equiv \frac{1}{2k}(T - T^{-1})$$

$$\nabla^2 : \frac{1}{h^2}\delta_h^2 + \frac{1}{k^2}\delta_k^2 \equiv \frac{1}{h^2}(S - 2I + S^{-1}) + \frac{1}{k^2}(T - 2I + T^{-1})$$

$$\nabla^4 : \left(\frac{1}{h^2}\delta_h^2 + \frac{1}{k^2}\delta_k^2\right)^2 \equiv \frac{1}{h^4}\delta_h^4 + \frac{2}{h^2k^2}\delta_h^2\delta_k^2 + \frac{1}{k^4}\delta_k^4$$

$$\equiv \frac{1}{h^4}(S^2 - 4S + 6I - 4S^{-1} + S^{-2}) + \frac{1}{k^4}(T^2 - 4T + 6I - 4T^{-1} + T^{-2}) \tag{4.8}$$

$$+ \frac{2}{h^2k^2}(ST - 2S - 2T + ST^{-1} + 4I + S^{-1}T - 2S^{-1} - 2T^{-1} + S^{-1}T^{-1})$$

$$\nabla^2\frac{\partial}{\partial x} : \left(\frac{1}{h^2}\delta_h^2 + \frac{1}{k^2}\delta_k^2\right)\frac{1}{2h}\bar{\delta}_h \equiv \frac{1}{2h^3}(S^2 - 2S + 2S^{-1} - S^{-2})$$

$$+ \frac{1}{2hk^2}(ST - 2S + ST^{-1} - TS^{-1} + 2S^{-1} - S^{-1}T^{-1})$$

$$\nabla^2\frac{\partial}{\partial y} : \left(\frac{1}{h_2}\delta_h^2 + \frac{1}{k^2}\delta_k^2\right)\frac{1}{2k}\bar{\delta}_k \equiv \frac{1}{2k^3}(T^2 - 2T + 2T^{-1} - T^{-2})$$

$$+ \frac{1}{2kh^2}(ST - 2T + S^{-1}T - ST^{-1} + 2T^{-1} - S^{-1}T^{-1})$$

Corresponding to the operators $\mathcal{N}$ and $\mathcal{L}$, these replacements generate a nonlinear difference operator $\mathcal{N}$ and a linear difference operator $\mathcal{L}$ given by

$$N(F_{ij}) \equiv \frac{1}{R}\left(\frac{1}{h^4}\delta_h^4 + \frac{2}{h^2k^2}\delta_h^2\delta_k^2 + \frac{1}{k^4}\delta_k^4\right)F_{ij} \tag{4.9}$$
$$- \frac{1}{4hk}\bar{\delta}_k F_{ij}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_h + \frac{1}{k^2}\delta_k^2\bar{\delta}_h\right)F_{ij}$$
$$+ \frac{1}{4hk}\bar{\delta}_h F_{ij}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_k + \frac{1}{k^2}\delta_k^2\bar{\delta}_k\right)F_{ij}$$

and

$$L(F_{ij}) \equiv \frac{1}{R}\left(\frac{1}{h^4}\delta_h^4 + \frac{2}{h^2k^2}\delta_h^2\delta_k^2 + \frac{1}{k^4}\delta_k^4\right) \tag{4.10}$$
$$+ \frac{1}{4hk}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_k + \frac{1}{k^2}\delta_k^2\bar{\delta}_k\right)F_{ij}\bar{\delta}_h - \frac{1}{4hk}\bar{\delta}_k F_{ij}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_h + \frac{1}{k^2}\delta_k^2\bar{\delta}_h\right)$$
$$- \frac{1}{4hk}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_h + \frac{1}{k^2}\delta_k^2\bar{\delta}_h\right)F_{ij}\bar{\delta}_k + \frac{1}{4hk}\bar{\delta}_h F_{ij}\left(\frac{1}{h^2}\delta_h^2\bar{\delta}_h + \frac{1}{k^2}\delta_k^2\bar{\delta}_h\right)$$

In terms of these the difference equations for the ϕ_{ij} read

$$L(F_{ij})\phi_{ij} = -N(F_{ij}) \tag{4.11}$$

Using the explicit expressions in (4.8) for the operators occurring in (4.10), we can readily find the coefficients multiplying the $\phi_{i+\mu,j+\nu}$ in (4.11), where $\mu,\nu = 0, \pm1, \pm2$ with $|\mu| + |\nu| \leq 2$. Denoting these coefficients by α_μ^ν (see Fig. 2), so that (4.11) reads

$$\sum_{|\mu|+|\nu|\leq 2} \alpha_\mu^\nu \phi_{i+\mu,j+\nu} = -N(F_{ij}), \tag{4.12}$$

and defining for convenience

$$a_1 = \frac{1}{Rh^4}, \quad a_2 = \frac{4}{Rh^2}\left(\frac{1}{h^2} + \frac{1}{k^2}\right), \quad a_3 = \frac{2}{Rh^2k^2}, \quad a_4 = \frac{4}{Rk^2}\left(\frac{1}{h^2} + \frac{1}{k^2}\right), \quad a_5 = \frac{1}{Rk^4},$$

$$b_1 = \frac{1}{4h^3k}\bar{\delta}_k F, \quad b_2 = \frac{1}{4h^3k}(\delta_h^2\bar{\delta}_k F) + \frac{1}{4k^3h}(\delta_k^2\bar{\delta}_k F), \quad c_2 = \frac{1}{2hk}\left(\frac{1}{h^2} + \frac{1}{k^2}\right)\bar{\delta}_k F, \tag{4.13}$$

$$c_3 = \frac{1}{4h^3k}\bar{\delta}_h F, \quad b_3 = \frac{1}{4k^3h}\bar{\delta}_k F, \quad b_4 = \frac{1}{4h^3k}(\delta_h^2\bar{\delta}_h F) + \frac{1}{4k^3h}(\delta_k^2\bar{\delta}_h F),$$

$$c_4 = \frac{1}{2hk}\left(\frac{1}{h^2} + \frac{1}{k^2}\right)\bar{\delta}_h F, \quad b_5 = \frac{1}{4k^3h}\bar{\delta}_h F$$

we find that:

$$\alpha_0^0 = a_2 + a_4 + 2(a_1 + a_5),$$

$$\begin{aligned}
&\alpha_{-2}^0 = a_1 + b_1, \quad \alpha_{-1}^0 = -a_2 - b_2 - c_2, \quad \alpha_{-1}^{-1} = a_3 + b_3 - c_3, \\
&\alpha_2^0 = a_1 - b_1, \quad \alpha_1^0 = -a_2 + b_2 + c_2, \quad \alpha_1^{-1} = a_3 - b_3 - c_3, \\
&\alpha_0^{-2} = a_5 - b_5, \quad \alpha_0^{-1} = -a_4 + b_4 + c_4, \quad \alpha_{-1}^{1} = a_3 + b_3 + c_3, \\
&\alpha_0^2 = a_5 + b_5, \quad \alpha_0^1 = -a_4 - b_4 - c_4, \quad \alpha_1^1 = a_3 - b_3 + c_3
\end{aligned} \tag{4.14}$$

With this notation we also find that

$$N(F_{ij}) = a_1\delta_h^4 F_{ij} + a_3\delta_h^2\delta_k^2 F_{ij} + a_5\delta_k^4 F_{ij} - b_2\bar{\delta}_h F_{ij} + b_4\bar{\delta}_k F_{ij} \tag{4.15}$$

To complete the system of equations for the ϕ_{ij} we must adjoin to the system (4.12) the equations corresponding to the boundary conditions. These will be considered in the next section.

4.3 Boundary Conditions

Normally two boundary conditions will be imposed along each of the four sides of the basic rectangular domain defined by the four grid lines with $i = 1, m$, $j = 1, n$. We will discuss several relevant pairs of conditions, but for simplicity will restrict attention to one boundary only, the one along the grid line $i = m$. The corresponding details for the other three sides will follow in an obvious way. Certain differences have to be observed between the $i = 1, m$ and the $j = 1, n$ boundaries when we consider the solution of the difference equations, but these need not concern us at present.

The most common pair of boundary conditions is for ψ and its normal derivative $\partial\psi/\partial x$ to be prescribed functions of the tangential variable y.

If the initial guess is set up with the correct boundary values of ψ, the first condition is simply represented by the equation

$$\phi_{m,j} = 0 \quad (j = 1, \ldots, n) \tag{4.16}$$

If the same standard difference approximation as in Section 4.2 is used, the second condition can be represented by

$$\psi_{m+1,j} = q + \psi_{m-1,j} \tag{4.17}$$

where $q = 2h(\partial\psi/\partial x)|_{i=m}$, or in Newton form with $\psi = F + \phi$:

$$-\phi_{m-1,j} + \phi_{m+1,j} = -F_{m+1,j} + F_{m-1,j} + q \tag{4.18}$$

Now this condition involves a grid value outside of the rectangle of stored values, which must be eliminated by using the appropriate stream-function equation written for one point in from the boundary. Here this reads

$$\ldots + \alpha_0^0\phi_{m-1,j} + \alpha_1^0\phi_{m,j} + \alpha_2^0\phi_{m+1,j} = -\alpha_2^0 F_{m+1,j} + \ldots \tag{4.19}$$

where only the relevant terms have been specifically written down. The remaining terms on the right-hand side do not involve $F_{m+1,j}$. Elimination of $\phi_{m+1,j}$ yields

$$\ldots + (\alpha_0^0 + \alpha_2^0)\phi_{m-1,j} + \alpha_1^0\phi_{m,j} = -\alpha_2^0(q + F_{m-1,j}) + \ldots \tag{4.20}$$

Thus the second boundary condition can be accounted for by modifying the Newton form of the stream-function equation for grid points one line in from the boundary. This modification amounts to replacing α_0^0 by $\hat{\alpha}_0^0 = \alpha_0^0 + \alpha_2^0$ and evaluating the right-hand side in the standard way by (4.15), except that wherever $F_{m+1,j}$ would occur, we use in its place $q + F_{m-1,j}$.

If the second normal derivative is given instead of the first, (4.17) will be replaced by

$$\psi_{m+1,j} = q + 2\psi_{m,j} - \psi_{m-1,j} \tag{4.21}$$

where now $q = h^2(\partial^2\psi/\partial x^2)|_{i=m}$, and the corresponding Newton form reads

$$\phi_{m-1,j} - 2\phi_{m,j} + \phi_{m+1,j} = -F_{m+1,j} + 2F_{m,j} - F_{m-1,j} + q \tag{4.22}$$

Again we must eliminate $\phi_{m+1,j}$ now between (4.19) and (4.22); the modified equation corresponding to (4.20) reads

$$\ldots + (\alpha_0^o - \alpha_2^o)\phi_{m-1,j} + (\alpha_1^o + 2\alpha_2^o)\phi_{m,j} = -\alpha_2^o(q + 2F_{m,j} - F_{m-1,j}) + \ldots \tag{4.23}$$

So again the boundary condition is accounted for by modifying the Newton form of the stream-function equation at the appropriate grid point and by evaluating the right-hand side using (4.21) with ψ replaced by F wherever $F_{m+1,j}$ occurs.

It may also be relevant to impose two boundary conditions involving $\psi_{m+1,j}$. For example, a satisfactory way of dealing with the downstream boundary condition in developing channel flow is to impose $\partial\psi/\partial x = \partial^2\psi/\partial x^2 = 0$ at stations suitably far downstream. If these stations are on the grid line $i = m$, we have

$$\psi_{m+1,j} - \psi_{m-1,j} = \psi_{m+1,j} - 2\psi_{m,j} + \psi_{m-1,j} = 0 \tag{4.24}$$

Eliminating $\psi_{m+1,j}$ between the two equations, we can replace the second by

$$\psi_{m,j} = \psi_{m-1,j} \tag{4.25}$$

which in Newton form reads

$$-\phi_{m-1,j} + \phi_{m,j} = F_{m-1,j} - F_{m,j} \tag{4.26}$$

We can now merely add this equation in place of (4.16) and account for the first of (4.24) by (4.18) with $q = 0$.

It may be noted that the pairs of conditions so far considered can all be included as special cases if we assume that the pairs of boundary difference equations can be put in the form

$$F_{m+1,j} = q + aF_{m,j} + bF_{m-1,j}, \qquad F_{m,j} = r + cF_{m-1,j} + dF_{m-2,j} \tag{4.27}$$

It will be seen in the next section that equations of this form do not disrupt the band structure of the Newton matrix; in fact, we will see that the first of the pair could also involve $F_{m-2,j}$ and $F_{m-3,j}$ without disrupting the structure, but this is not necessary for the cases we consider.

Note that the first equation of (4.27) is applied for $j = 2, \ldots, n-1$, i.e. at $n-2$ points, and hence, totalling the corresponding equations for the other sides and also the Newton equations for the stream function, we have $MN-4$ equations without those corresponding to the second equation of (4.27). The latter must, therefore, total $2M + 2N-4$, since the total number of unknowns is $MN + 2(M-2) + 2(N-2)$. Thus, at each corner we must choose just one of the associated sides to contribute a boundary condition corresponding to the second of (4.27). In most applications it will not matter which, so we choose arbitrarily to associate the corners with the North and South sides ($j = 1, n$) (see Fig. 2).

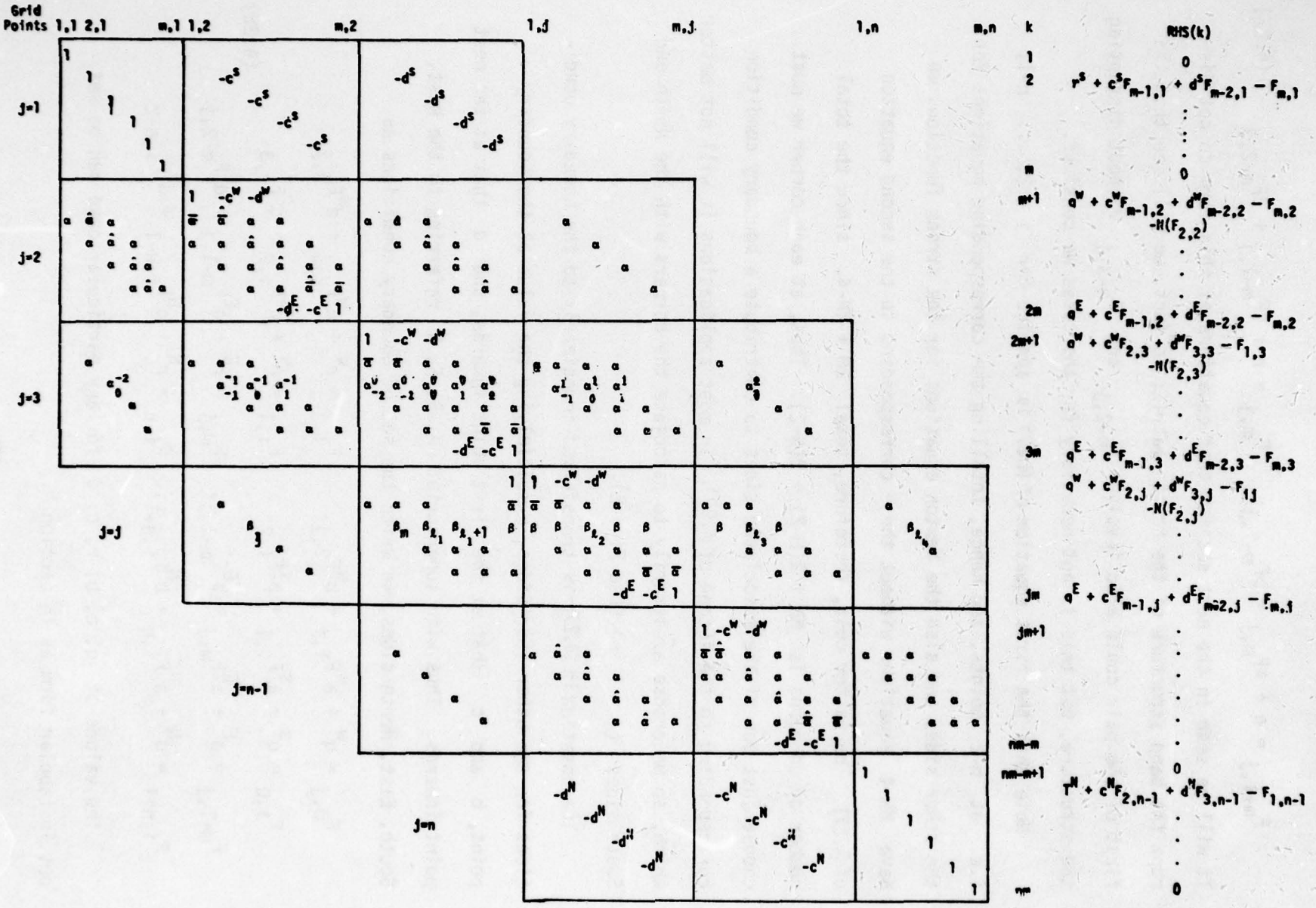
The notation in (4.27) is transferred immediately to the boundary conditions for the other four sides, a multiplying the value at the boundary point, b and c that at the first point inwards, and d that at the next point inwards. Thus with superscript W, S, E, N referring to the West, South, East, North sides, we write the set of boundary conditions as

$$\begin{aligned}
F_{0,j} &= q^W + a^W F_{1,j} + b^W F_{2,j}, & F_{1,j} &= r^W + c^W F_{2,j} + d^W F_{3,j} \\
F_{i,0} &= q^S + a^S F_{i,1} + b^S F_{i,2}, & F_{i,1} &= r^S + c^S F_{i,2} + d^S F_{i,3} \\
F_{m+1,j} &= q^E + a^E F_{m,j} + b^E F_{m-1,j}, & F_{m,j} &= r^E + c^E F_{m-1,j} + d^E F_{m-2,j} \\
F_{i,n+1} &= q^N + a^N F_{i,n} + b^N F_{i,n-1}, & F_{i,n} &= r^N + c^N F_{i,n-1} + d^N F_{i,n-2}
\end{aligned} \tag{4.28}$$

The values of q, a, b, r, c, d for any particular case can be set out in tabular form as in section 5.

Figure 3.

NEWTON MATRIX FOR BIHARMONIC FORMULATION



4.4 Matrix Structure and Solution by Band Solver

There are various ways available for the solution of the linear system (4.11). Several block elimination schemes can be formulated and also iterative schemes, including those of ADI type. Since one would like to take advantage of the Chord or special Newton method, a scheme in which an LU decomposition is performed would seem to be preferable. Block elimination schemes can be designed to do this, but because of its potentially greater stability, especially for large Reynolds numbers, we choose to explore the practical application of a standard band solver with partial pivoting for stability (this may be especially important since we hope to work entirely in IBM single precision).

The standard way of organizing the system (4.11) as a banded system is to order the unknowns by rows, i.e. in the order $\phi_{1,1}, \cdots, \phi_{m,1}, \phi_{1,2}, \cdots$ The matrix elements for the system (4.11) plus the boundary equations of the general form (4.28) has the appearance shown in Fig. 3. The superscripts N, S, E, W indicate to which of the North, South, East, West sides the second boundary conditions belong. For clarity the suffixes attached to the α's are shown on one row only, and the modifications to the α's due to eliminating the first boundary equations are only indicated by position. Thus, those needing modification at the W and E boundaries are shown with a bar, those at the N and S by a hat, and those at both by a bar and hat.

For use in the band solver, the diagonals have to be stored as the columns of a rectangular array, B say, so they are numbered from 1 to $\ell_4 = 4m + 1$. The main diagonals of the individual blocks have salient locations, which are denoted by ℓ_1, ℓ_2 and ℓ_3. These are shown in a sample row in Fig. 3, which also shows an equation count k in relation to the successive blocks.

It is important that the rows are scaled so that their norms are of fairly uniform size before the LU subroutine is called. The natural scaling

of equations (4.11) is good enough since the diagonal element α_0^0 is constant and is likely to be the dominant element in each row. However, the rows with the unit diagonal elements are normally much smaller, so scaling is definitely necessary, especially for single precision, as tests have indicated, and it is convenient to multiply them through by α_0^0. In some problems these unit diagonal elements do not arise if the known boundary values of ψ are not included as unknowns.

The major storage requirements are for the matrix B, whose dimensions are (mn, ℓ_4) and a matrix C, say, for storing the L of the LU decomposition, whose dimensions are (mn, ℓ_2), where $\ell_2 = 2m+1$. In addition, we need three vectors of dimension mn for F, the right-hand sides and the interchange permutation. This yields mn (6m+5) as an estimate for the storage requirements. Note that the solve procedure SOL after the decomposition needs both B and C, so that the Chord method needs the same amount of store as the full Newton iteration.

Operation counts yield the following asymptotic estimates for the cost of each stage of an iteration step:

$$LU \simeq 8M^3N, \qquad SOL \simeq 6M^2N, \qquad COEFFS \simeq 15MN$$

Here LU is the procedure for factoring L(F) into the LU form, and COEFFS denotes the procedure for evaluating the Newton matrix and right-hand sides. Thus, neglecting the cost of COEFFS, the ratio of a full Newton to a Chord step is about 4M/3 and as M increases it would seem more efficient to switch to the chord method. However, this depends on the rate of convergence of the linear iteration. This, in turn, depends on the closeness of the initial guess from which the Jacobian matrix is calculated. If a sequence of Reynolds number cases are performed, with or without extrapolation from one to the next, this will in turn depend on the Reynolds number spacing. Other

strategies can be envisaged, for example, two full Newtons could be used each step with chord iterates, if necessary, either between or after. It is not really clear what an optimum strategy might be since this is rather problem-dependent and also depends on the accuracy required. For the applications considered here we choose a fixed absolute accuracy of four decimal places in the stream function as a reasonable objective. Since the velocities are obtained by dividing stream function values by the step lengths, this corresponds roughly to graphical accuracy in the velocities. The vorticity can vary considerably in magnitude, so it would seem reasonable to allow the accuracy in the vorticity to be defined implicitly by the accuracy in the stream function.

5.0 TESTS AND COMPARISONS

A series of tests and comparisons of the basic methods has been made and is continuing. Considerable experience has been gained in designing iteration strategies for good convergence and it has been clearly demonstrated that the techniques developed avoid the convergence problems associated with large mesh Reynolds numbers. It should be pointed out, however, that the present restriction to uniform grid codes does put a severe limitation on the accuracy that can be obtained for large Reynolds numbers. A subsequent investigation will examine the extent to which this situation can be improved by introducing a variable grid of an appropriate character. In the present report we concentrate on convergence and consider two regimes. The first covers a Reynolds number range for which we may expect accurate results by h^2-extrapolation. The second deals with higher Reynolds numbers for which the grids that we can reasonably use are not fine enough for h^2-extrapolation to be of much significance, but for which convergence is still possible, because we avoid the mesh Reynolds number problem, and results can be obtained that give a reasonable qualitative picture. The introduction of a variable grid should give this regime more qualitative significance and also extend it.

The basic test problems are those of the driven cavity and entrance flow in a channel. The former problem has been used as a test by almost all workers in this area and thus is essentially mandatory. In addition, it isolates a basic feature which occurs in more complicated flows that we wish to consider later. The channel flow problem is perhaps closer to some of the practical problems we wish to study, and it also contains a free "downstream" boundary on which the development of appropriate "soft" boundary conditions can be studied.

5.1 Driven Cavity

We consider the biharmonic formulation first. Here the boundary conditions are very simply dealt with since the fact that $u = v = 0$ on all sides except one, say the North, where $u = 1$ and $v = 0$, implies that the coefficients in the boundary equations in (4.27) are all constants and mostly zero. In terms of ψ we have $\psi = \psi_x = 0$ on the East and West sides, $\psi = \psi_y = 0$ on the South side and $\psi = 0$, $\psi_y = 1$ on the North side. In terms of the net function F_{ij}, the conditions on the North side, for example, read

$$F_{i,n+1} = 2\Delta y + F_{i,n-1}, \qquad F_{i,n} = 0$$

so that $r^N = c^N = d^N = 0$, $q^N = 2\Delta y$, $a^N = 0$, $b^N = 1$. A similar treatment of the normal derivatives on the other sides yields the table

Side	q	a	b	r	c	d
W	0	0	1	0	0	0
S	0	0	1	0	0	0
E	0	0	1	0	0	0
N	$2\Delta y$	0	1	0	0	0

A wide range of Reynolds numbers from 25 to 2000 were tried on a very coarse 11×11 grid with a variety of iteration strategies to gain experience with the convergence properties. Several strategies produced convergence for $R = 1000$, but none of those used did so for $R = 2000$. It did not seem worthwhile exhausting all possibilities in order to see whether $R = 2000$ was an upper bound for this grid, but it does seem likely. In any case, although the results for $R = 1000$ gave a qualitative picture that was not unreasonable, this Reynolds number is well beyond that for which one could expect reasonably accurate results with a coarse uniform grid.

Several finer grids were tried and since 31 x 31 seemed to be a reasonable upper limit on the IBM 370 in the present context, most of the tests were performed on 11 x 11, 21 x 21, and 31 x 31 grids. Tests showed that the sequence R = 50, 100, 200 was a reasonable range to consider for the regime in which h^2-extrapolation could be expected to give graphical accuracy in the u velocity on the centerline x = 1/2, and the results are graphed in Fig. 4. The curves for R = 50, 100 are extrapolated from 21 x 21 and 31 x 31 and seem adequate for graphical accuracy. For R = 200 an extrapolation on all three has been performed and even this is probably not quite adequate for graphical accuracy; the curve for 31 x 31 is shown for comparison. A distinctive structure is beginning to appear at this Reynolds number. Results for R = 500 on the 31 x 31 grid are also shown; they indicate that the sharpening up of the structure is being followed, but an extrapolated curve is not given since errors are likely to be substantial to graphical accuracy.

Data for various iteration strategies are given in the following table.

	Iterations Required R =				Convergence Factors R =			Timings (370 mins)			
Grid	50	100	200	4M/3	50	100	200	Newton	1 Chord	Total	Average
11 x 11	1 + 5	9	16	15	0.26	0.59	0.77	0.010	0.001	0.31	0.010
21 x 21	1 + 7	11	27	28	0.41	0.64	0.81	0.138	0.003	0.314	0.105
31 x 31	1 + 7	12	33	41	0.44	0.67	0.85	0.625	0.012	1.270	0.423
31 x 31	1 + 7	1 + 2	1 + 4	41	0.44	0.07	0.53	0.625	0.012	2.061	0.687
31 x 31	2 + 3	1 + 2	1 + 5	41	0.39	0.14	0.80	0.625	0.012	2.656	0.885

The number of iterations is given in the form $i_1 + i_2$, or i_2 if $i_1 = 0$, where i_1 is the number of Newton iterations and i_2 the number of chord iterations. The general strategy was to specify two integers n_1 and n_2, and to perform a maximum of n_1 Newtons for the first Reynolds number case

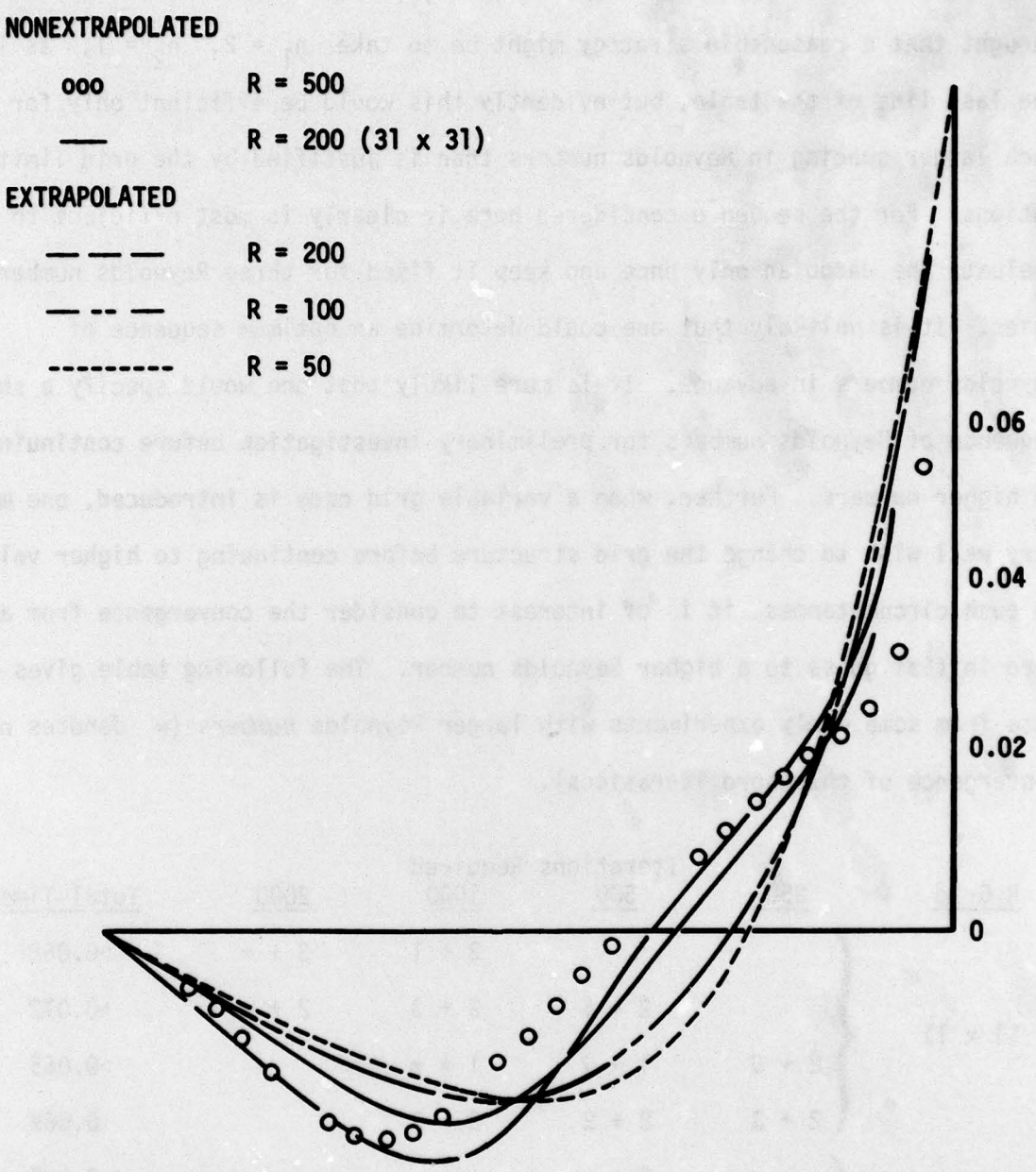


Fig. 4. Extrapolated curves of u on vertical centerline for R = 50, 100, 200

and a maximum of n_2 Newtons for the remaining ones, followed in all cases by chord iterations until convergence (max $|\phi| < \frac{1}{2}\, 10^{-4}$). At first it was thought that a reasonable strategy might be to take $n_1 = 2$, $n_2 = 1$, as in the last line of the table, but evidently this would be efficient only for a much larger spacing in Reynolds numbers than is justified by the grid limitations. For the sequence considered here it clearly is most efficient to evaluate the Jacobian only once and keep it fixed for three Reynolds number cases. It is unlikely that one could determine an optimum sequence of Reynolds numbers in advance. It is more likely that one would specify a short sequence of Reynolds numbers for preliminary investigation before continuing to higher numbers. Further, when a variable grid code is introduced, one may very well wish to change the grid structure before continuing to higher values. In such circumstances, it is of interest to consider the convergence from a zero initial guess to a higher Reynolds number. The following table gives some data from some early experiments with larger Reynolds numbers (∞ denotes non-convergence of the chord iterations).

R Grid	Iterations Required 250	500	1000	2000	Total Time
11 x 11			3 + 1	3 + ∞	>0.068
		2 + 5	2 + 3	2 + ∞	>0.072
	2 + 3	1 + 9	1 + ∞		>0.065
	2 + 3	2 + 2	2 + 3		0.069
21 x 21		2 + ∞			>0.335
		3 + 11			0.447
	3 + 3				0.412
	2 + 8				0.302
31 x 31		3 + ∞			>2.209
	3 + 5	2 + 2			3.178
	3 + 5	1 + 8			2.669

It would be needlessly expensive to make this table more complete, but some observations may be made from selected items. In going straight to $R = 500$, for example, we see that more Newtons may be needed as the grid is refined; and for the sequence $R = 250, 500, 1000$ with the 11×11 grid the strategy $n_1 = 2$, $n_2 = 1$ ultimately fails for a Reynolds number for which a solution exists.

The comparison with the $\psi - \zeta$ ADI scheme of section 3 is strictly a comparison between methods for solving the difference equations since the difference approximations are identical. What we are comparing is a $\psi - \zeta$ iteration scheme in which the ζ equations are solved by ADI and the ψ equations are solved by SOR with a biharmonic scheme in which the iteration matrix is factorized with a big band solver. The first thing that can be said is that the biharmonic scheme needs vastly more storage than the $\psi - \zeta$ scheme, at least $mn\,(6m + 5)$ words, which for a 31×31 grid is approaching 2×10^5, as compared with little more than $6mn$, which could be reduced still further if the convenience of storing u and v was relinquished. Clearly one must look for a compensating advantage in the biharmonic scheme. The boundary conditions are somewhat more easily dealt with in the biharmonic scheme, but this is fairly marginal. The obvious remaining characteristic is speed or run time to achieve a given accuracy. This is not quite so easy to compare as might be expected, since the $\psi - \zeta$ iterations can be very slowly convergent, in which case the same tolerance test $\max |\Delta\psi| < \frac{1}{2} 10^{-4}$ would not give a fair comparison. This is because the actual error can be many times larger than the iterative change in cases of very slow linear convergence. If, as usually the case, the convergence is ultimately geometric with a common ratio less than but close to unity, a comparison with the geometric series $s_n = a + ar + \ldots + ar, n-1$ for which $|S_n - \lim S_n| = |S_n - S_{n-1}| \times r/(1-r)$, suggests that we should terminate the iterations

when $\max |\Delta\psi| < \frac{1}{2}\, 10^{-4} \times (1-r)/r$, where r is an estimate of the common ratio of the size of the successive corrections. Since the actual ratio of successive corrections will not necessarily be monotonic and smooth, we use the spatial average of the $|\Delta\psi|$ and, in fact, store the last eight values of this quantity, say $a_1, a_2, \ldots, a_8$, and take r as given by $r^4 = (a_5 + a_6 + a_7 + a_8)/(a_1 + a_2 + a_3 + a_4)$. Comparison of several sample values of the converged results from each method shows that they give exactly the same results for the stream function to the four decimal places required, and also that, if the original criterion of $\max |\Delta\psi| < \frac{1}{2}\, 10^{-4}$ is used, substantial differences in the fourth decimal do occur.

A further complication to a straightforward comparison is the choice of pseudo time step, for experiments show that for each grid and each R there is an optimum Δt for which convergence is reached in the smallest number of steps. Unless we use this optimum value, there is no definite scheme which we can say is typical. For the present this has been found by varying the parameter $\gamma = \Delta t/\Delta x$; for example, $\gamma \approx 4.3$ for $R = 100$ with the 31 x 31 grid and thus yields a value of $r \approx 0.92$ for the asymptotic convergence factor. It seems that r is less sensitive to γ for $\gamma < \gamma_{opt}$ than for $\gamma > \gamma_{opt}$. In fact, r increases quite rapidly for $\gamma > \gamma_{opt}$, and it may be possible to use this to estimate γ_{opt} by increasing γ as the solution progresses until the currently monitored estimate of r exceeds 1.0 and then backing off a notch. For the typical case mentioned above the time to convergence (in 84 iterations) is about 0.125 compared with 0.625 for 1 Newton on a 31 x 31 grid, so there is a fair amount of leeway that could be taken up in such monitorings. One could also afford to store several earlier iterations and carry out some form of extrapolation such as Aitken's.

Thus in the Reynolds number regime where h^2-extrapolation is accurate, the ADI scheme seems to be substantially faster. One presumes that the convergence will get slower as the Reynolds number increases and that the problem of finding an optimum γ will worsen. This turns out to be the case as the table below indicates, but even for $R = 500$ the time required is still less than 1 Newton on the 31 x 31 grid. However, considerable time was used in experimenting to find γ_{opt} and attempts at automating this have so far not been satisfactory. Total time is therefore becoming quite comparable with the biharmonic scheme.

R	100	250	500
Optimum γ	4.3	4.5	4.7
Convergence factor	0.92	0.96	0.98
Iterations required	84	220	400
Time required (370 mins)	0.125	0.322	0.587

There would seem to be more point in developing the variable grid codes before trying to push the Reynolds number up further. For variable grid codes more time would be spent in evaluating the residuals, which would tend to improve the speed ratio between the biharmonic and the $\psi - \zeta$ schemes, so one may expect the biharmonic scheme to be preferable with a variable grid, especially at Reynolds numbers in excess of 500.

5.2 Channel Flow with Uniform Parallel Inlet Velocity

We make use of symmetry and take the channel wall to be along $x = 0$ and the centerline along $x = \frac{1}{2}$. Entry conditions are $u = 0$, $v = 1$ along $y = 0$ and fully developed flow is assumed to have been reached by $y = y_{max}$. Normally y_{max} is very much bigger than $\frac{1}{2}$, so we expect to have to take more stations in the downstream direction than across the channel. An appropriate y_{max} will depend on Reynolds number and in fact should be proportional to R

for large R. Experiments showed that $y_{max} = 3.0$ was reasonable for R = 50 so for larger values we took $y_{max} = 0.06R$. This turned out to be excessive for larger R, presumably because we are not in the asymptotic regime at R = 50.

A downstream boundary condition in which the fully developed profile was imposed generated oscillatory behavior near the downstream boundary, so a condition of parallel flow independent of y was imposed instead. This avoided the oscillations and yielded a downstream profile adequately close to the known parabolic profile. Thus, the boundary conditions read:

$x = 0$: $u = 0$, $v = 0$, $\quad x = \frac{1}{2}$: $u = 0$, $v_x = 0$

$y = 0$: $u = 0$, $v = 1$, $\quad y = y_{max}$: $u = 0$, $u_y = 0$

or in terms of ψ:

$x = 0$: $\psi = 0$, $\psi_x = 0$, $\quad x = \frac{1}{2}$: $\psi = -\frac{1}{2}$, $\psi_{xx} = 0$

$y = 0$: $\psi = -x$, $\psi_y = 0$ $\quad y = y_{max}$: $\psi_y = \psi_{yy} = 0$

Using standard central difference approximations and the eliminations mentioned in 4.3, we obtain the following table

Side	q	a	b	s	c	d
W	0	0	1	0	0	0
S	0	0	1	-x	0	0
E	0	2	-1	$-\frac{1}{2}$	0	0
N	0	0	1	0	1	0

For a selection of moderate Reynolds numbers, including R = 150 for which we can make comparisons with the results of Wang and Longwall[12], we obtain the following performance data, starting from a first initial guess given by fully-developed flow.

Grid	370 Timings (minutes) 1 Newton	1 Chord	Total	Average	R =	Iterations Required 50	100	150	200	Convergence Factors 50	100	150	200
11 x 31	0.031	0.0057	0.156	0.051		1 + 13	23		30*	0.51	0.86		0.95
			0.181	0.060		2 + 8	25		30*	0.45	0.87		0.96
			0.135	0.045		1 + 13	1 + 3		1 + 3	0.51	0.07		0.21
			0.098	0.098				3 + 2				0.07	
21 x 61	0.397	0.0126	1.704	0.568		1 + 29*	1 + 4		1 + 6	0.82	0.45		0.50
			1.147	diverged		2 + 28*				1.14			
			1.715	0.858		3 + 4		1 + 6		0.42		0.80	
			1.202	1.242				3 + 4		0.31			

Results for the case $R = 150$ on the 21 x 61 grid with $y_{max} = 9.0$, $\Delta x = 0.025$, $\Delta y = 0.15$, obtained with the biharmonic program, were compared with the results of Wang and Longwall[12]. Satisfactory agreement was obtained. A further series of tests on a crude 8 x 15 grid were made to investigate the convergence for larger Reynolds number. The strategy defined by $n_1 = 2$, $n_2 = 1$ worked extremely well on the Reynolds number sequence $R = 25, 50, 100, 200, 400, 800, 1000, 3200$, the convergence for $R = 3200$ appearing to be just as good as for the earlier values.

A limited number of tests have been made with the $\psi - \zeta$ ADI program for this case because it soon became evident that the biharmonic scheme was clearly superior and a lot of pointless work would have been carried out in finding γ_{opt}. Suffice it to say that a typical particular case, $R = 50$ on a 11 x 31 grid, converged in about 230 iterations with $\gamma = 5.0$ and convergence factor 0.97, and took 0.15 mins, as compared with 0.11 mins for the same single case using the biharmonic program.

6.0 REFERENCES

1. Keller, H. B. and Cebeci, T.: Primitive Variable Methods for the Navier-Stokes Equations. MDC Rept. J7891, March 1978.
2. Patankar, S.V. and Spalding, D.B.: A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows. Int. J. Heat Mass Transfer, Vol. 15, pp. 1787-1806, 1972.
3. Chorin, A.J.: On the Convergence of Discrete Approximations to the Navier-Stokes Equations. Math. of Computation, Vol. 23, No. 106, pp. 341-353, 1969.
4. Amsden, A.A. and Harlow, F.H.: The SMAC Method: A Numerical Technique for Calculationg Incompressible Fluid Flows. LA-4370, Los Alamos Scientific Lab., Los Alamos, N.M., 1970.
5. Lilley, D.G.: Primitive Pressure-Velocity Code for the Computation of Strongly Swirling Flows. AIAA J., Vol. 14, No. 6, pp. 749-756, 1976.
6. Pope, S.B.: Private Communication.
7. Roache, P.J.: Computational Fluid Dynamics, Mermosa Publishers, Albuquerque, N.M,, 1972.
8. Carlo, A.D., Piva, R. and Gui, G.: Numerically Mapped Macro-Elements for Multiply Connected Flow Fields. Paper presented at the 6th-International Conf. on Numerical Meth. in Fluid Mechanics, Tbilisi, USSR, 1978.
9. Anon: Numerical Studies of Incompressible Viscous Flow in a Driven Cavity, NASA SP-378, 1975.
10. Peaceman, D.W. and Rachford, H.H., Jr.: The Numerical Solution of Parabolic and Elliptic Differential Equations. SIAM J. on Applied Mathematics, Vol. 3, No. 1, pp. 28-41, 1935.
11. Roache, P.J.: The LAD, NØS and SPLIT NØS Methods for the Steady State Navier-Stokes Equations. Computers and Fluids, Vol. 3, pp. 179-195, 1975.

12. Wang, Y.L. and Langwell, P.A.: Laminar Flow in the Inlet Section of Parallel Plates. A.I.Ch.E.J., Vol. 10, No. 3, pp. 323-329, 1964.